

TARTU ÜLIKOOL
SOTSIAAL- JA HARIDUSTEADUSKOND

Haridusteaduste instituut
Haridusteaduste (reaalained) õppekava

Kai Sellin

**RoboMind kui võimalus äratada põhikooli õpetajate ja õpilaste
huvi programmeerimise vastu**
bakalaureusetöö

Juhendaja: Eno Tõnisson
Kaasjuhendaja: Sirje Pihlap

Läbiv pealkiri: Programmeerimise töölehtede katsetamine

KAITSMISELE LUBATUD

Juhendaja: Eno Tõnisson (MSc)

.....

Kaasjuhendaja: Sirje Pihlap

.....

Kaitsmiskomisjoni esimees: Liina Lepp (MA)

.....

Tartu 2014

SISUKORD

SISUKORD	2
SISSEJUHATUS	4
I TAUST	7
1.1 Informaatikaõpetusest ja programmeerimise õpetamise suundadest Eesti üldhariduskoolides	7
1.2 <i>Computational thinking</i> oskus ja programmeerimise õpetamise tähtsus põhihariduse seisukohalt	9
1.3 Lühike ülevaade suundadest teiste riikide informaatikaõpetuses	11
1.4 Lühike ülevaade lastele loodud programmeerimise õppimise vahenditest	12
1.5 Programmi <i>RoboMind</i> lühitutvustus	16
II METOODIKA	18
2.1 Õppematerjal	18
2.2 Uuritavad ja protseduur	19
2.3 Mõõtevahendid	22
III TULEMUSED	23
3.1 Tulemused õpetajate tagasiside põhjal	23
3.1.1 Õpetajate eelnev programmeerimise õpetamise ja õppimise kogemus	23
3.1.2 Õpetajate seisukohad programmeerimise õpetamise suhtes üldhariduskoolides. ...	23
3.1.3 Õpetajate hinnangud töölehtedele ja programmile	26
3.2 Tulemused õpilaste tagasiside põhjal	26
3.2.1 Õpilaste eelnevad teadmised programmeerimise kohta.	26
3.1.2 Õpilaste suhtumine programmeerimise õppimisse.	27
3.2.3 Õpilaste hinnangud töölehtedele ja programmile	29

IV ARUTELU	31
Kokkuvõte	35
Summary	36
Autorsuse kinnitus	37
Kasutatud kirjandus	38
Lisad	42
Lisa 1. Esimene pöördumine õpetajate poole (kaaskiri nr. 1)	42
Lisa 2. Teine pöördumine õpetajate poole (kaaskiri nr. 2)	44
Lisa 3. Elektrooniline ankeet õpetajatele	46
Lisa 4. Elektrooniline ankeet õpilastele	55
Lisa 5. Töölehtede komplekt I	59
Lisa 6. Töölehtede komplekt II	65
Lisa 7. Õpetajakoolitustel osalenud õpetajate geograafiline paiknemine Eestis	66

SISSEJUHATUS

Vilipõld, Antoi ja Amitan (2013) leiavad, et üksnes info- ja kommunikatsioonitehnoloogia (IKT) vahendite kasutamise oskusest jääb tänastel koolilõpetajatel ja tulevastel spetsialistidel väheseks. Nad toovad välja, et lisaks info kasutamise ja loomise oskusele, tuleks tööle või ülikooli õppima asumisel kasuks ka infosüsteemide ja tarkvara loomise, arendamise ja haldamise ning programmjuhtimisega süsteemide arhitektuurialased teadmised. Selliste oskuste järkjärguline areng võiks kuuluda kooliinformaatika õpiväljundite juurde. Hetkel tegeletakse Eesti üldhariduskoolides nende väljatoodud teadmiste ja oskuste arendamisega kas osaliselt või üldse mitte. Nimelt on Eestis põhikoolides õpilastel enamasti võimalus õppida IKT-vahendeid kasutama (Põhikooli riiklik õppekava, 2011, lisa 10, pt 1.1.1 p 1), kuid seeläbi pööratakse tähelepanu eespool toodud info kasutamise ja loomise oskusele. Gümnaasium pakub mitmes valikaines õpilastele võimalust omandada informaatikaalaseid teadmisi süvendatumalt (Gümnaasiumi riiklik õppekava, 2011, lisa 4). Samas on nii algsete kui põhjalikumate informaatika- ja ka IT-alaste teadmiste omandamine hetkel üldhariduskoolides valikuline. Riigi tasandil pole määratud, et need teadmised peaksid kohustuslikus korras jõudma kõigi Eesti koolide õpilasteni.

Võib öelda, et õpilased on tänasel päeval teistsuguste harjumuste ja maailmapildiga kui aastakümneid tagasi. Romeike (2008) toob välja, erinevate tehnoloogiliste vahendite kasutamine muutub järjest igapäevasemaks tänapäeva laste elus. Samas leiavad Resnick jt (2009), et õpilased kasutavad IT-vahendeid info otsimiseks, mängimiseks või suhtlemiseks, kuid vähesed oskavad neid vahendeid millegi loomiseks või leiutamiseks kasutada. Hromkovič ja Steffen (2011) märgivad, et informaatika võib pakkuda laialdast võimalust probleeme püstitada, neile lahendusi otsida ja midagi ise üles ehitada (näiteks programmide loomise näol). Kelleher (2009) märgib, et informaatikaalased teadmised ja oskused muutuvad kõrghariduse ja ka tööle minemise seisukohast järjest olulisemaks. Õpilasi nii Eestis kui välismaal suunatakse järjest enam ettevõtlikkust üles näitama ja oma ideedele rakendust otsima (vt JA Eesti õpilasfirma). Ka suuremad informaatikaalased teadmised võiksid õpilastele ideede realiseerimisel abiks olla. Euroopa Komisjoni Innovatsioonikonverentsil erinevatest riikidest osalenud õpilastest ettevõtjad tõid oma ütlustes välja, et tahaksid koolis tegeleda rohkem individuaalsete projektidega ning õppida arvutis keerulisemaid teemasid kui näiteks Microsoft Wordiga töötamine (vt Äripäev). Eestis on uuritud õpilaste arusaamu IKT

valdkonnast ja nende rahulolu informaatikaõpetusega uuringu „*IKT hariduse populaarsus külastatud koolide õpilaste seas (2012)*“ käigus. Suur osa seal uuringus osalenud 9.-12. klasside õpilasi leiab, et infotehnoloogia eriala on keeruline. Uurimistulemustest selgus, et kõige rohkem ollakse informaatikatunnis rahul arvutiklassiga ning kõige vähem õpetuse sisuga. Tüdrukutega võrreldes olid poisid nii arvutiõpetuse sisu kui ka arvutiklassiga vähem rahul. Reaalainetes viielised ja nelja-viielised õpilased olid samuti õpetuse sisuga vähem rahul. Uurijad nendivad, et koolide arvutiõpetuse tase ei ole hetkel piisavalt hea. (vt IKT hariduse populaarsus..., 2012) Hetkel pole selge, kuidas mõjutab informaatikaõpetuse tase koolides õpilaste edasisi haridusalaseid valikuid ja tööalast edu. Kuid võib öelda, et informaatikaõpetuse tase ja sisu vajavad jätkuvalt haridusedendajate tähelepanu ja mõtestamist.

Käesolevas bakalaureusetöös uuritakse sellise informaatikaalase oskuse nagu programmeerimise õpetamise võimalusi üldhariduskoolide reaalainete tundides, kuid eelkõige informaatikatundides. Reaalainete tundide all mõistetakse siinkohal matemaatika-, füüsika- ja informaatikatunde. Resnick jt (2009) nendivad, et üldsuse silmis on programmeerimine ainult väga kitsale spetsialistide ringile mõeldud tegevus. Antud töös püütakse selgitada programmeerimise võimalusi ja kasulikkust õpilaste mõtlemise arengule ka üldhariduse seisukohalt. Programmeerimine kuulub eelnevas lõigus mainitud tarkvara loomise, arendamise ja haldamise oskuste juurde, kuid samas annab see tegevus teatud ettevalmistuse mõistmaks ka programmjuhitavate süsteemide ülesehitust ja toimimist. Programmjuhitavate süsteemidega puutuvad inimesed igapäevaselt järjest rohkem kokku. Sealjuures arendab programmeerimise õppimine õpilastel loogilist mõtlemist ja analüüsivõimet ning pakub rohkelt võimalusi ise midagi luua, katsetada, eksida, kuid tänu kiirele tagasisidele ka kohe vigu otsida ja parandada. Resnick jt (2009) toovad välja, et kui arvutid aastakümneid tagasi suuremal määral tavatarbijani hakkasid jõudma, siis loodeti, et kõik lapsed hakkavad mingil hetkel programmeerimist õppima. Autorid nendivad aga, et tollel hetkel ei olnud selliseid programmeerimiskeeli, mis oleksid võimaldanud noortele huvipakkuvalt ja kerge vaevaga programmeerimist tutvustada ja seetõttu kestis vaimustus koolides lühikest aega. Praegu on olemas mitmeid selliseid keskkondi, kus programmeerimise õppimine võiks laste jaoks olla huvitav, lõbus ja eduelamust pakkuv. Selliste keskkondade alla kuulub autori meelest ka antud uurimistöö raames töölehtede loomise aluseks olnud programm *RoboMind*. Programmi valimise põhimõtetest ja selle ülesehitusest tuleb hiljem rohkem juttu.

Arvestades põhjalikumate informaatikaalaste teadmiste omandamise kasulikku mõju õpilaste mõtlemisele ajal, mil informaatikaõpetuse tase ei pruugi alati vastata ei nende ega ka

üldisemalt ühiskonna nõudmistele, on oluline uurimisprobleem, kuidas koolis programmeerimist õpetada. Üle maailma on mitmete spetsialistide poolt loodud võimalusi, et ka väga noored õpilased võiksid programmeerimisega mingil määral tegeleda. Käesolev bakalaureusetöö on uudne selle poolest, et varem pole autorile teadaolevalt Eestis uuritud selle kooliastme õpilaste ja nende õpetajate valmidust ja huvi reaalinete tundides programmeerimisega tegeleda.

Antud töös seati eesmärgiks koostada Eesti põhikooliklasside õpilastele hinnanguliselt 1-2 koolitunniks programmi *RoboMind* abil programmeerimise õppimiseks vajalikud töölehed ja saada pärast tundide läbiviimist nendele ja programmeerimise õppimise võimalusele üldisemalt õpilaste ja õpetajate poolt tagasisidet. Töö käigus koostati kaks ankeeti, mille abil sooviti saada vastused järgmistele uurimisküsimustele:

1. Kuidas suhtuvad õpetajad programmeerimise õpetamisesse põhikoolis?
2. Kuidas võiks programmeerimist õpetajate arvates põhikooli tasemel õpetada?
3. Kuidas hindavad õpetajad koostatud töölehtede sobivust programmeerimise õpetamiseks?
4. Kuidas hindavad õpetajad valitud programmi *RoboMind* sobivust programmeerimise õpetamiseks?
5. Kuidas suhtuvad õpilased programmeerimise võimalikku õppimisse põhikoolis?
6. Kuidas hindavad õpilased koostatud töölehtede huvitavust ja keerulisust?
7. Kuidas hindavad õpilased valitud programmi *RoboMind*?

Antud töö koosneb kahest osast. Esimeses osas arutletakse üldhariduskoolide informaatikaõpetuse sisu ja programmeerimise õpetamise suundade üle nii Eestis kui mõnes teises Euroopa riigis. Samuti antakse ülevaade programmeerimisel kasutatavatest oskustest, selle õppimise kasulikest omadustest ja mõnedest programmeerimise õppimise vahenditest. Esimese osa lõpus tutvustatakse lühidalt ka antud töölehtede koostamise aluseks olnud programmi *RoboMind*. Teises osas kirjeldatakse töölehtede ja ankeetide koostamise põhimõtteid, antakse ülevaade uuritavatest ning töölehtede ja ankeetide uuritavatele edastamise protseduurist. Samuti kirjeldatakse teises osas saadud tulemusi ning arutletakse nende, vajalike muutuste ja edasise uurimise võimaluste üle.

I TAUST

1.1 Informaatikaõpetusest ja programmeerimise õpetamise suundadest Eesti üldhariduskoolides

Selles peatükis tuleb juttu informaatikaõpetuse sisust ja eesmärkidest ning programmeerimise õpetamise suundadest ja huvist nende vastu Eesti koolides. Samuti tuuakse peatüki lõpus lühike ülevaade põhjustest, miks üldiselt koolis oleks kasulik programmeerimisega tegeleda.

Eestis õpivad põhikooli õpilased arvutitega töötama erinevates ainetundides, kuid suuremalt jaolt tähendab see töö info hankimist, selle töötlemist ja esitlemist. Põhikoolis on paljudel õpilastel võimalus nende oskuste omandamisele keskenduda valikaines *Informaatika*. Laanpere (2010a) toob välja, et ühiskonnas on aastaid olnud surve koolidele ja haridusedendajatele, et informaatikaõpetus muutuks põhikooli tasemel kohustuslikuks kõikides koolides, kuid eelkõige on takistuseks olnud koolide arvutipark ja kvalifitseeritud õpetajate puudus. Micheuz (2011) võrdleb informaatikaõpetust näiteks matemaatika või inglise keelega ja leiab, et nende puhul on tegemist pikemat aega viimistletud õppekavadega, samas kui informaatika näol on tegemist üsna uue õppeainega koolis.

Informaatikaõpetuse üldeesmärk Põhikooli riikliku õppekava (2011) kohaselt on tagada IKT-vahendite rakendamise pädevused eelkõige koolis tekkivaid nõudmisi arvestades. Valikaines *Informaatika* ei ole programmeerimise õpetamine sätestatud. Reaalteaduste õppesuunaga koolidele soovitatakse informaatikatunnis sissejuhatuse tegemist *arvutiteadusesse*¹, kuid missugune see sissejuhatus võiks olla ja kas see hõlmaks ka programmeerimise õpetamist, pole samuti sätestatud. Laanpere (2010a) märgib, et kui ainekavas ettenähtud teemad suudetakse informaatikatunnis kiiresti läbida, siis on õpetajal vabadus õpetada õpilastele ka näiteks programmeerimist spetsiaalselt noorematele õpilastele mõeldud *Scratchi* (antakse põhjalikum ülevaade pt. 1.4) või *Lego Mindstorms*i abil.

Võib öelda, et koolides on viimaste aastakümnete jooksul erineval ajal, viisil ja tasemel programmeerimisega tegeletud. Laanpere (2010b) toob välja, et kooliinformaatika õpetamise eesmärki on seostatud digitaalse kirjaoskuse kujundamisega, mis kunagi tähendas programmeerimiseoskust, kuid tänapäeval on suuremalt jaolt meedia- ja suhtluspädevuse arendamisega seotud. Samas leidub Eestis ka praegu koole, kus püütakse informaatikatunnis,

¹ Siinkohal tuleb manida, et mõistet *arvutiteadus* võidakse kasutada erinevates allikates erinevas tähenduses. Antud töös on kasutatud selle asemel ja mõiste *computer science* tõlkimisel mõistet *informaatika*.

huviringis või valikkursusel üldhariduskoolide õpilastele programmeerimist õpetada, näideteks Gustav Adolfi Gümnaasium (vt Kes õpetab..., 2014), Tallinna Reaalkool (vt Tallinna Reaalkool), TTÜ Tehnoloogiakool (vt TTÜ Tehnoloogiakool) ja Tartu Erakool (vt Erahuvikool HuviTERA). Olulisel määral tegeletakse programmeerimisega Eesti koolirobootika projekti raames, millega on liitunud 81 kooli üle Eesti (vt Kooliroboti projekt). Õpetajate koolitamisele pööratakse järjest suuremat tähelepanu Hariduse Infotehnoloogia Sihtasutuse (HITSA) Innovatsiooni keskuse poolt (vt HITSA). HITSA poolt finantseeritava *ProgeTiigri* programmi (vt programm ProgeTiiger) raames toimuvad kursused õpetajatele ja huviringide juhendajatele, mis baseeruvad programmeerimiskeeltele *Scratch* ja *Python*. Kaardi sellest, kuidas jagunevad õpetajakoolitusest osa võtnud õpetajad Eestis, võib leida Lisas 7.

Laanpere (2010b) leiab, et arvutiprogrammid, mida koolides kasutama õpitakse, on muutunud järjest lihtsamalt mõistetavaks ja kasutatavaks. Romeike (2008) lisab, et ka operatsioonisüsteemid on muutumas järjest kasutajasõbralikumaks ning kuna õpilased oskavad üldiselt arvutit kasutada, võiks informaatikatunnis arvuti kasutamise õpetamise asemel keskenduda rohkem informaatikale. Sealjuures nendib Romeike (2008), et informaatika tuumaks on probleemide lahendamine ning selliste ülesannete lahendamine on ka programmeerimise õpetamisele omane. Laanpere (2010b) soovitab Eesti õpetajatel olla informaatika õpetamisel loomingulised ning nendib, et IKT teemade ja üldiselt informaatika õpetamine peab ajaga kaasas käima.

Programmeerimist ei pea aga ilmtingimata informaatikatunnis õpetama, sest seda saab põhimõtteliselt teha ka näiteks matemaatika- või füüsikatunnis. Programmeerimise õpetamine ei pea olema ainult ühe aine keskne, vaid võib olla kasulik ka teiste ainete tundides. Seda just seetõttu, et programmeerimise ülesannete lahendamine aitab kaasa õpilaste mõtlemisoskuse arenemisele erineval tasemel. Lastele mõeldud programmeerimise keskkonnad toetavad suures osas õpilaste mängulisust ja fantaasiat. *ProgeTiigri* programmi raames toimuvatele kursustele on oodatud kõik õpetajad, kes soovivad. Mitmed autorid on oma töödes välja toonud, kuidas programmeerimine on kasulik ka teiste reaalainetundide õppetöös. Calder (2010) viitab Scratchiga programmeerimise kasulikule mõjule matemaatilisele mõtlemisele. Sengupta jt (2013) arutlevad programmeerimise ja computational thinking oskuse rakendamise võimaluste üle USA koolide *STEM (Science, Technology, Engineering and Math)* õppekavades. Programmeerimise abil matemaatiliste teemade õppimise võimalustele on viitavad ka Kahn, Sendova, Sacrista'n ja Noss (2011). Kindlasti aitaks programmeerimise õppimine kaasa ka näiteks õpilaste inglise keele oskusele, kuna suur osa haridusliku suunitlusega programmeerimiskeeli on ingliskeelsed. Isegi kui õppekava ainete tasemel

programmeerimist ei sätesta, siis peaks programmeerimise eesmärgistatud õpetamine kaasa aitama järgmistele põhikooli riiklikus õppekavas toodud alusväärtustele ning õppe- ja kasvatusesmärgidele:

- §2(1) (...) Põhikool loob tingimused õpilaste erisuguste võimete tasakaalustatud arenguks ja eneseteostuseks ning teaduspõhise maailmapildi kujunemiseks.
- §2(4) Uue põlvkonna sotsialiseerumine rajaneb eesti kultuuri traditsioonide, Euroopa ühisväärtuste ning maailma kultuuri ja teaduse põhisaavutuste omaksvõtul. (..)
- §3(2) Põhikoolis on õpetuse ja kasvatuses põhitaotlus tagada õpilase eakohane tunnetuslik, kõlbeline, füüsiline ja sotsiaalne areng ning tervikliku maailmapildi kujunemine.
- §3(3) Põhikooli ülesanne on luua õpilasele eakohane, turvaline, positiivselt mõjuv ja arendav õppekeskkond, mis toetab tema õpihimu ja õpioskuste, eneserefleksiooni ja kriitilise mõtlemisvõime, teadmiste ja tahteliste omaduste arengut, loovat eneseväljendust ning sotsiaalse ja kultuurilise identiteedi kujunemist.

(Põhikooli riiklik õppekava, 2011)

Olgugi et programmeerimise õpetamine võiks olla huvitav ja kasulik ka teiste reaalainete tundides, seostatakse antud töös ka edaspidi programmeerimise õpetamist eelkõige informaatikaga.

1.2 *Computational thinking*² oskus ja programmeerimise õpetamise tähtsus põhihariduse seisukohalt

Selles peatükis tuleb juttu mõiste *computational thinking* tähendusest ja seosest programmeerimisega ning tuuakse välja põhjused, miks oleks kasulik põhihariduse tasemel tähelepanu pöörata sellise oskuse arendamisele ning seda näiteks läbi programmeerimise õppimise.

Denner ja Werner (s.a.) märgivad, et esimest korda kirjeldas mõistet *computational thinking* Seymour Papert 1993. aastal ja Jeannette Wing võttis 2006. aastal selle mõiste uuesti kasutusele. Denning (2009) leiab, et *computational thinking* on sama, mis 1950ndatel tunduks saanud *algorithmic thinking*. Veelgi enam, ta rõhutab, et *computational thinking* on oluline teaduse arengu seisukohast ja see idee iseenesest on pärit teistest teadustest, mitte arvutiteadusest. Mõiste *computational thinking* võetakse siin töös kasutusele aga järgneval

² Tuleb märkida, et püüdlused seda mõistet tõlkida ei andnud tulemusi ning autor ei võta siinkohal vastutust, et anda sellele mõistele endapoolne ja seega ebapädev tõlgendus.

põhjusel. Kuuendal rahvusvahelisel kooliinformaatika konverentsil 2013. aastal (*6th International Conference on Informatics in Schools: „Situation, Evolution, and Perspectives“*), kus osalesid nii teadlased kui õpetajad, esitleti uuenduslikke töid kooliinformaatika õpetamisteemadel. Nendel konverentsidel on aastaid kestnud arutelu, mida ja kuidas informaatikatunnis õpetada. Kui mõnda aega on eri riikide koolides tegeletud põhiliselt e-õppe tutvustamise ja erinevate IKT-vahendite kasutamise õpetamisega, siis viimastel aastatel on jõutud küsimuseni, kuidas arendada õpilastes *computational thinkingut*. (Diethelm & Mittermeir, 2013)

Vilipõld, Antoi ja Amitan (2013) märgivad, et viimasel ajal on USAs haridusedendajate seas hakanud levima mõiste *computational thinking*, millele eestikeelne vaste veel puudub. Vilipõld jt leiavad, et seda mõistet saab kasutada sealhulgas ka selliste arendustöö elementide puhul nagu modelleerimine, algoritmimine ja programmeerimine. Google'i veebilehel *Exploring Computational Thinking*, millele Vilipõld jt viitavad, seletatakse see mõiste lahti järgnevalt: „*Computational thinking* hõlmab ülesannete lahendamise oskusi ja meetodeid, mida kasutavad tarkvara insenerid, et kirjutada programme, mis on aluseks arvuti rakendustele (...) nagu näiteks otsingumootorid, e-post ja kaardid. See oskus on rakendatav aga peaaegu igal erialal.“ Seega programmeerijad kasutavad *computational thinking* oskust oma töös pidevalt. *Computational thinking* tähendab oskust arutleda (Henderson, Cortina, Hazzan, Wing, 2007) ja oskust analüütiliselt mõelda (Wing, 2008). *Computational reasoning* (*reasoning – põhjendamine*) kui osa *computational thinkingust* on tänapäeva teaduse, tehnoloogia, tehnika ja matemaatika distsipliinide tuumaks ja inimene kasutab seda pidevalt igapäevaselt tegutsedes, näiteks kasvõi süüa tehes (Henderson jt, 2007).

Wing (2008) näeb *computational thinking* oskuses lähenemisviisi, kuidas tegeleda matemaatiliste probleemide lahendamisega, projekteerida erinevaid keerukaid süsteeme, mis toimiksid reaalses elus ja mõista inimkäitumist- ja intellekti. Ta leiab, et see puudutab kõiki otseselt või kaudselt leides kasutust igasugustes ettevõtmistes erinevatel erialadel tasemel nii täna kui ka tulevikus. Seega tõstatab ta küsimuse, kuidas ja millal peaks seda õpetama. Wing (2008) loodab, et ülikoolides on *computational thinking* leidnud juba tee õppekavadesse ning asetab rõhu kooliharidusele. Ta leiab, et järgmine generatsioon peab oskama nii mõelda selleks, et edukalt toimida. Kwiatkowska ja Sysło (2013) on veendunud, et *computational thinking* õppimine võiks koolis olla võrdse tähtsusega lugema, kirjutama ja arvutama õppimisega. Nad rõhutavad, et just keskkooliõpilased, kes hakkavad valmistuma erialaliseks spetsialiseerumiseks, peaksid oskama arvutusmeetodeid- või vahendeid õigesti kasutada.

Programmeerimist õppetöösse kaasates oleks võimalik õpilastel arendada ka *algorithmic* ehk *computational thinking* oskust. Ioannidou, Bennett ja Repenning (2011) rõhutavad, et informaatikas arvutuslikku soravust (*computational fluency*) taotledes peaks lisaks programmeerimise oskusele tähelepanu pöörama ka teistele vajalikele oskustele (näiteks mängude kujundamisega seotud küsimused). Nad toovad välja, et üldsuse seas levib väärarusaam, justkui informaatika ja programmeerimine oleks üksühesed ja seega on tihti õpilaste ning eriti tüdrukute huvi informaatika vastu madal. Diethelm ja Mittermeir (2013) rõhutavad, et kooliinformaatikas ei tohiks kalduda ka teise äärmusesse, mille tulemusena õpilastele tundub, et neid koolitatakse programmeerijateks. Põhikoolis ei ole mõistlik programmeerimist õpetada sellisel tasemel või eesmärgil nagu teatud õppekavadel ülikoolides või rakenduskõrgkoolides, kuid kindlasti on võimalik kõrgkoolide kogemusest õppida.

Diethelm ja Mittermeir (2013) märgivad, et põhiküsimus seisneb siiski õpetajate koolituses. Toimetajad leiavad, et tihti puudub informaatikaõpetajatel vajalik koolitus, et õpilastega tõsisemaid informaatika ülesandeid sealhulgas programmeerimise ülesandeid lahendada. Siegle (2009) märgib, et enamikul arvutikasutajatest ei pruugi kunagi tekkida vajadust programmeerida, kuid nendib samas, et paljudele õpilastele see tegevus meeldib. Ta leiab, et programmeerimine nõuab sealhulgas planeerimis- ja korraldamisoskust ning täpsust ja enesedistsipliini. Mitmed autorid leiavad, et programmeerimise õppimisega võiksid kokku puutuda õpilased juba põhikoolis. Repenning (2012) märgib, et programmeerimist õpetatakse mõningatele õpilastele koolivälise tegevuse raames. Samas rõhutab ta välja, et juba põhikooliõpilased võivad võtta omaks näiteks sellise suhtmise, et *reaalained pole nende jaoks*. Ta püstitab küsimuse, kuidas tõsta põhikooli informaatika hariduse tähtsust nii, et seda õpetatakse nõutud tasemel kõigis koolides. Kelleher (2009) märgib, et eduelamuse pakkumine juba põhikooli õpilastele programmeerimise esmasel õppimisel võib tuua rohkem õpilasi kõrgemal tasemel informaatika alast haridusteed jätkama, kuid nendib, et tulevikus läheb informaatika alaseid oskusi vaja ka muudel teadusaladel õppides ja töötades.

1.3 Lühike ülevaade suundadest teiste riikide informaatikaõpetuses

Eelpool mainitud rahvusvahelise kooliinformaatika konverentsi materjalide hulgast on valitud kolme autori tööd, mille põhjal antakse järgnevalt ülevaade olukorrast kolme Euroopa riigi informaatikaõpetuses sealsetes üldhariduskoolides. Antud riigid valiti näideteks seetõttu, et valitud tööde autorite meelest vajab kooliinformaatika sisu nendes riikides uuendamist ja kaasajastamist.

Inglismaal, kus viimase kahekümnendi jooksul on informaatika õpetuses rõhku pandud info- ja kommunikatsioonitehnoloogia (IKT) vahendite rakendamise õpetamisele, toimvad nüüd informaatika õppekavas kiired muutused. Erinevate programmide kasutamisoskuse kõrval ei ole praeguses õppekavas üldse tähelepanu pööratud programmeerimis- ja *computational thinking* oskusele. Sealne asutus *Computing At School* (CAS), mis tegeleb informaatika hariduse edendamise ja informaatikaõpetajate toetamisega, koos kõrgharidusest ja tööstusest lähtuva survega on olnud toimuvate muutuste aluseks. Muutustele aitas kaasa ka *Google*'i tegevjuhi Eric Schmidti kriitika, et õpilasi õpetatakse kasutama erinevat tarkvara, aga see-eest ei ole neil ettekujutust, kuidas see tarkvara ise on tekkinud. Sellest tulenevalt vaadatakse nüüd õppekava üle. Samuti koolitatakse õpetajaid, et tõsta nende teadmisi informaatikas ning luua võimalused, et tulevikus asuksid tööle mõned põhjalikumate informaatikaalaste teadmistega õpetajad. (Sentance, Dorling & McNicol, 2013)

Prantsusmaal ei õpetata informaatikat põhikoolis eraldi ainaena. Seal õpivad õpilased IKT-vahendeid kasutama teiste ainete tundides. Uurimused näitavad aga, et õpilastel puudub sügavam arusaamine informaatika mõistetest ja mehhanismidest. Nüüd vaieldakse, kas õpilased peaksid varakult ja süvendatumalt kokku puutuma informaatikaga. Põhjuseks tuuakse välja vajadus teadmiste järele, kuidas käituda digitaalses ühiskonnas ja soov suurendada tõenäosust, et õpilased jätkaksid õpinguid erinevates teadusvaldkondades. (Tort & Drot-Delange, 2013)

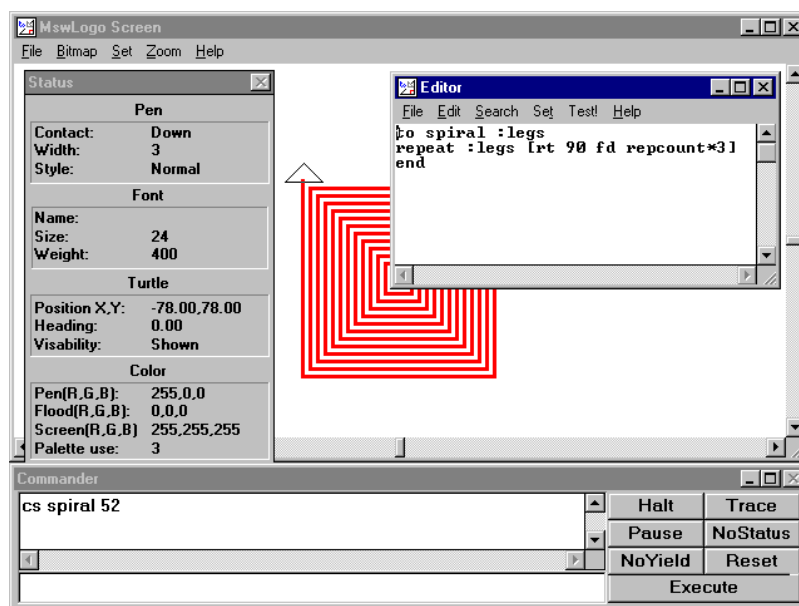
Poolas on läbi pidevate vaidluste püsinud informaatika riiklikus õppekavas alates 1985. aastast eraldi õpetatava ainaena, kus tegeletakse ka tõsisemate informaatika teemadega. Sealne põhikooli õppekava sisaldab muuhulgas ka algoritmikat, *algoritmilise mõtlemise* õppimist ja arvuti abil probleemide lahendamist. Mõnedes koolides tutvuvad õpilased ka selliste programmeerimiskeeltega nagu *LOGO* (antakse põhjalikum ülevaade pt. 1.4) või *Scratch*. Üldiselt esineb nende teemade käsitlemisel siiski koolides muresid, sest õpetajatel võib vähestest oskustest ja teadmistest tulenevalt esineda vähest enesekindlust õpetada teemasid õpilastele, kellel võib programmeerimises eelnevalt kogemusi olla. (Sysło & Kwiatkowska, 2013)

1.4 Lühike ülevaade lastele loodud programmeerimise õppimise vahenditest

Lastele mõeldud programmeerimise õppimise vahendeid uurides ilmneb, et valikuvõimalus on üsnagi suur. Programmeerimisvõimalusi võivad pakkuda sealhulgas ka erinevad mängud või veebilehed, mis ei ole hariduslikul eesmärgil koostatud. Leidub vähe

ülevaatlisku informatsiooni kõigi nende vahendite kohta ehk millised keskkonnad rohkem või vähem sobiksid erinevates vanuserühmades õpilastele programmeerimise õpetamiseks. Seega oleks olnud keeruline ka antud töös sellekohast ülevaadet anda. Resnick jt (2009) toovad välja, et tuhadetes koolides õpetati 1970ndatel ja 1980ndatel aastatel, kui arvutid rohkem levima hakkasid, õpilastele programmeerimist sellistes keeltes nagu *LOGO* ja *Basic*. Ühed tuntumad programmeerimiskeskkonnad, mille kohta on teaduslikke uurimusi läbi viidud, on *LOGO*, *Scratch* ja *Alice*. Järgnevalt antakse lühike ülevaade nendest kolmest lastele programmeerimise õppimiseks mõeldud keskkonnast.

LOGO arendati välja Seymour Paperti ja programmeerijate tiimi poolt 1967. aastal. *LOGO* loomisel olid olulised kaks tingimust, et sellega saaksid töötada algajad (sealhulgas lapsed), kuid et jääks võimalus tegeleda ka keerulisemate programmeerimise ülesannetega. *LOGO* tegelaskuju sai oma kuju järgi nimeks *kilpkonn* ning seda sai kontrollida lihtsate käskude abil nagu *edasi* ja *tagasi* või *pliiats alla*. Just see oligi lastega programmeerimise juures oluline, et oleks võimalik koheselt näha tulemust ja sellest õppida. (Ratcliff & Anderson, 2011) Joonisel 1 on kujutatud programmi *LOGO* aken.

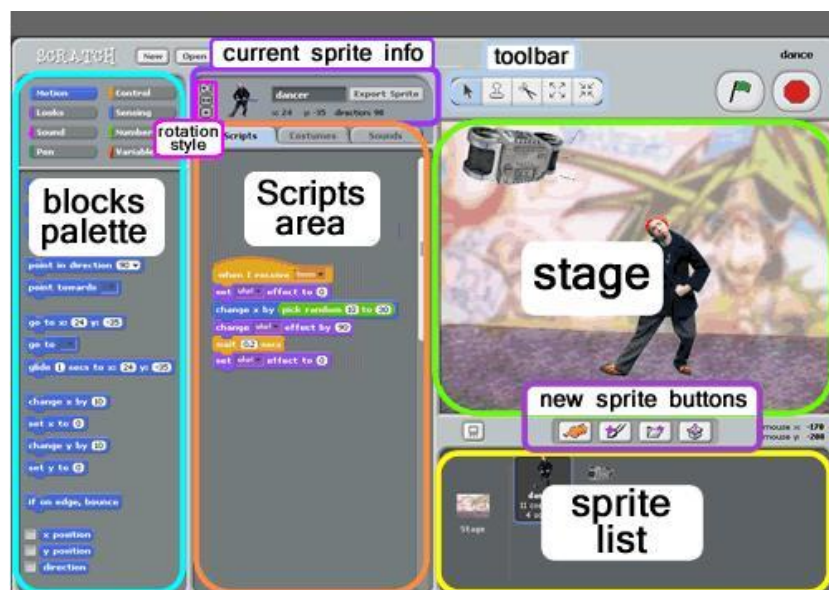


Joonis 1. Programmi *LOGO* aken, kus tegelaskuju *kilpkonn* on programmeeritud liikuma spiraalselt (<http://blogs.msdn.com/b/ukschools/archive/2011/08/25/who-would-have-thought-a-turtle-would-have-come-such-a-long-way.aspx>)

Üks programmi loojatest Feurzeig (2010) toob välja, et *LOGO* eesmärgiks oli arendada laste mõtlemisoskust nii, et nad läbi aktiivse tegevuse ise teadmisi looks. Ta märgib, et programm *LOGO* annab hea aluse õpilaste matemaatiliste teadmiste arenguks. Samuti leiab ta, et

*LOGO*ga programmeerimine arendab aineüleseid õpioskusi nagu näiteks enesehindamine, aja planeerimine ja enesepeegeldus. Võib öelda, et *LOGO* on üks vanimaid lastele programmeerimiseks loodud keeli ning ilmselt seetõttu üks tuntumaid. *LOGO* on olnud aluseks ka uuematele täiustunud programmeerimiskeskondadele nagu näiteks *GeoLogo*, *NetLogo* ja *Scratch*. Feurzeig (2010) lootis, et tänu arvutite ja nende rakenduste kiirele arengule leiavad programmeerimise ideed ja tegevused oma tee koolidesse, kuid nendib nüüd, et nende kasutus on pigem hääbumas ja potentsiaal siiski veel avastamata.

Scratchi ametlik versioon ilmus 2007. aastal, kuid selle projektiga alustati juba mitu aastat varem (vt *Scratch*). Lee (2011) võrdleb *Scratchi* teiste programmeerimiskeeltega ja näeb positiivsena, et *Scratchis* saab trükkimise asemel arvutihiire abil panna kokku nõ pusle tükke ja seega luua kergema vaevaga keerulisemaid programme. Autor toob *Scratchi* puhul veel välja, et sealsed tegevused hõlmavad sealhulgas ka animatsioonide, mängude ja taustamuusika loomist. Joonisel 2 on kujutatud programmi *Scratch* aken.

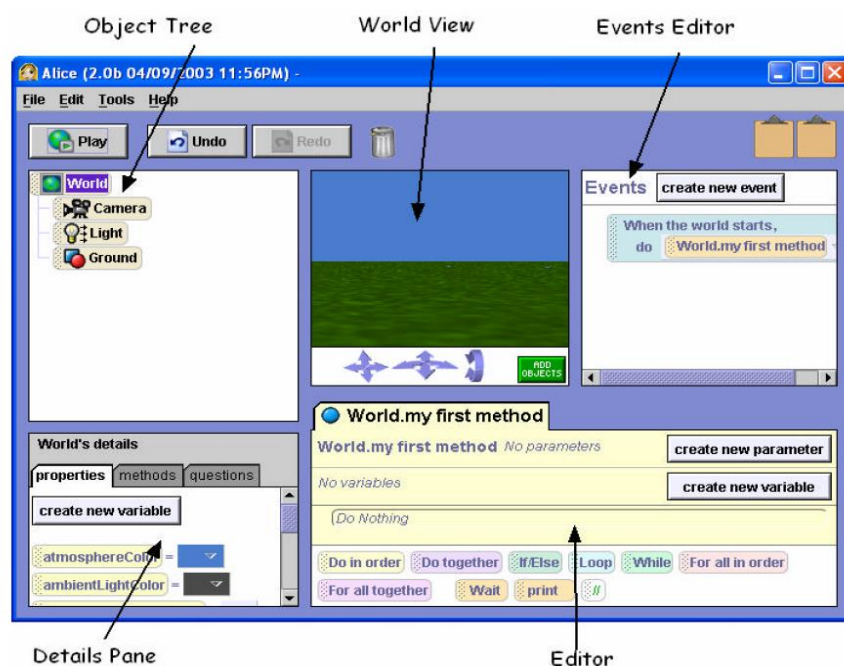


Joonis 2. Programmi *Scratch* aken, kus on inglise keeles info kasutajaliidese osade kohta. (http://scratched.media.mit.edu/sites/default/files/scratch_interface_1.JPG)

Resnick jt (2009) märgivad, et tänapäeva noored käivad digitaalse tehnoloogiaga osavalt ringi, kuid samas ei oska enamus neist ise näiteks mängu luua. Taoline digitaalne soravus hõlmaks endas loomise ja kavandamise oskust, mitte noorte seas juba levinud oskusi lehitseda ja vestelda. Calder (2010) märgib, et Scratchiga programmeerides teevad õpilased mõõtmisi, mis aitavad kaasa geomeetria alaste teadmiste arenemisele. Lee (2011) toob veel ühe

positiivse omadusena välja, et Scratchiga programmeerimist toetavad veebilehed sisaldavad abimaterjale nii lastele kui täiskasvanutele nii õppimiseks kui õpetamiseks.

Programmeerimiskeele *Alice* puhul on tegemist projektiga, mida on arendanud USA Carnegie Melloni Ülikooli tiim 1990ndatest aastatest alates (vt *Alice*). Selle projekti eesmärgiks on muuta lihtsaks algajatele programmeerimise õppijatele 3-D keskkondade arendamine³. 3-D objektideks seal on näiteks loomad. *Alice* on eelkõige skriptimise⁴ ja proovimudelite tegemise keskkond. *Skripte* kirjutades saavad kasutajad muuta objektide välimust või käitumist. *Alice* sobib hästi algajatele programmeerijatele just seetõttu, et kasutajatel on kohe visuaalselt võimalik näha, kuidas nende programmeeritud animatsioon töötab. Algajatel programmeerimise õppijatel on nimelt tihtilugu raske ette kujutada, kuidas arvuti nende poolt koostatud programmi täidab (*execution*) ning seega ka raske oma vigu parandada, kui miski peaks minema valesti. (Cooper, Dann & Pausch, 2000) Joonisel 3 on kujutatud programmi *Alice* aken.



Joonis 3. Programmi *Alice* aken, kus on inglise keeles märgitud kasutajaliidese osad.

(https://www.alicechallenge.org/image/Alice%20Interface%20Graphic.png/image_view_fullscreen)

Kelleher (2009) näeb *Alice* programmeerimiskeeles võimalust motiveerida põhikooli õpilasi ja iseäranis tüdrukuid õppima arvutiga programmeerimist. Ta näeb järjest suurenevat probleemi soolises lõhes informaatika õpingutes ning leiab, et positiivne kogemus üldhariduskoolides programmeerides aitaks suurendada huvi informaatika õppimise vastu.

³ Mõiste *arendamine* tuleneb mõistest *tarkvara arendustöö*, mida mainiti sissejuhatuse teises lõigus.

⁴ Skript kujutab endast käsujada. (IT-sõnastik)

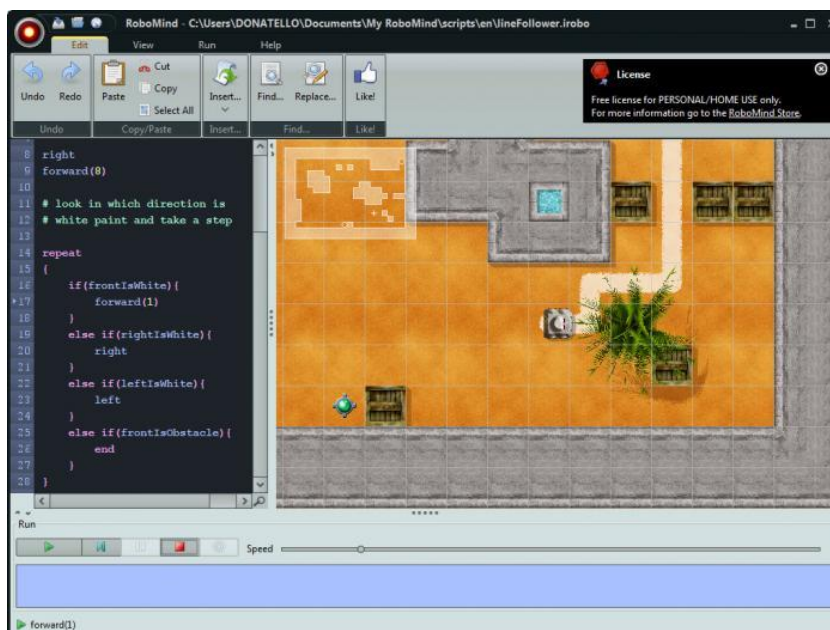
Siin töös lühidalt kirjeldatud programmeerimise õppimise vahendid ei ole ilmtingimata just kõige sobivamad igale klassile. Nagu eespool mainitud, siis kasutatakse Eestis õpetajakoolitustel näiteks *Scratch* ja *Python* keeli. Sobiva keele valik võib sõltuda õpilaste huvidest, vanusest, õpetaja silmaringist ja konkreetse vahendi võimalustest.

1.5 Programmi *RoboMind* lühitutvustus

Antud töö käigus 2012. aastal valiti õpilastele programmeerimise õppimise võimaluste tutvustamiseks programm *RoboMind*. Järgnevalt antakse lühike ülevaade programmist ja selle kasutamisevõimalustest ning tuuakse välja põhjused, miks just see programm valiti antud uurimuse läbiviimise jaoks.

RoboMind loodi 2005. aastal *Research Kitchenis* seal magistrandina õppinud Arvid Halma, elektriinsener Ernst Bovenkamp'i ja professor Jan van Oorschot'i poolt. Seda tutvustatakse kui suurepärase vahendit informaatika õppimiseks ja üldisemalt *computational thinking* oskuse ja loogilise mõtlemise arendamiseks. *RoboMind* sobib programmeerimise õppimiseks põhikoolist ülikoolini ning seda kasutatakse hetkel mitmetes koolides üle maailma. (vt *RoboMind*) Tuleb märkida, et võrreldes 2012. aastaga on *RoboMindi* kodulehte rikastatud erinevate ülesannete ja võimalustega. Seega võib öelda, et huvi selle programmi vastu on aja jooksul suurenenud. Enam ei ole tegu vabavaralise programmiga, vaid programmi kasutamise hind aastas sõltub sellest, kui palju lisavõimalusi kasutaja soovib ning kas soovitakse omandada ka *computational thinking* diplomeid. Hetkel on võimalik programmiga tasuta töötada 30 päeva. Tuleb märkida, et nüüdseks pakutakse *RoboMindi* kasutajatele ka võimalust katsetada kirjutatud programme *Lego Mindstorms* robotitega ehk *legorobotitega*.

Joonisel 4 on kujutatud programmi *RoboMind* aken. Akna vasakus pooles on võimalik programmeerida ning paremal asub keskkond, kus on võimalik näha programmeerimise tulemusi.


 Joonis 4. Programmi *RoboMind* aken.

Programmeerimiskeskonna tegelaskujuks on *robot*. Robotit saab juhtida nii käsurea kui puldi abil. Programmi avades saab kasutaja vihje, et esimesel kasutamisel võiks kõigepealt kasutada pulti. Robot on suuteline liikuma, pead pöörama, asju liigutama ja värvima (vt *RoboMind*). *RoboMindi* kasutajale pakutakse ka erinevaid kaarte või võimalust ise kaarte luua, kus *robotiga* toimetada. Käsud, mida kasutada programmeerimisel, võib leida *Insert* menüüst. Seega kasutajal on vajadusel ülevaade kõikidest võimalikest antud programmis programmeerimiseks kasutatavatest käskudest.

Siinkohal tuuakse välja põhjused, miks just antud tööks see programm hästi sobis. Nagu öeldud, sobib *RoboMind* kasutamiseks ka põhikooli õpilastele. Programm oli uurimuse läbiviimise hetkel tasuta ja lihtsasti kättesaadav. Programmi kasutajaliides on lihtsa ja loogilise struktuuriga ning jätab atraktiivse üldmulje. Algajate programmeerijate seisukohast on oluline, et *robotit* saab juhtida nii puldi kui programmeerimise abil. Kindlasti ei saa öelda, et kõik *RoboMindi* kasutamise juures on alati hästi. Töölehti oli keeruline ülesannetega täiendada, sest tasuta versioonis on toimetamiseks ainult üks kaart, mida pole võimalik näiteks ise muuta. Samuti piirab tegutsemisvabadust ka see, et robot alustab igat käsku ühest ja samas punktist. Olgugi et nüüdseks hetkeks on *RoboMindi* kasutamine muutunud tasuliseks, on sealjuures ka programmi võimalused laienenud. Seega võib öelda, et *RoboMindi* poolt pakutavate võimaluste abil soovitakse laste hulka, kes programmeerimist õpiksid, suurendada.

II METOODIKA

Antud töö käigus koostati õppematerjal programmeerimise õppimiseks ning anti see õpetajatele ja õpilastele katsetamiseks ja hindamiseks. Järgnevalt antakse ülevaade õppematerjali loomise, selle edastamise ja katsetamise tingimustest.

2.1 Õppematerjal

Uurimuse läbiviimiseks koostati hinnanguliselt kuni kaheks koolitunniks õppematerjal. Õppematerjaliks on töölehed, mis on mõeldud programmiga *RoboMind* õppimiseks. Töölehtede komplektid võib leida Lisades 5 ja 6. Õppematerjal loodi autori enda poolt ja ei kasutatud näiteks mõne teise uurimuse õppematerjali seetõttu, et valitud keskkond *RoboMind* sobis hästi põhikooli õpilastele programmeerimises algeliste teadmiste omandamiseks, kuid puudusid selles keskkonnas toimetamiseks eestikeelsed toetavad ja suunavad töölehed. Õppematerjali koostamisel eeldati, et ei õpilastel ega õpetajatel pole selles valdkonnas varasemaid kogemusi.

Õppematerjal on jaotatud kaheks osaks. I tunni materjal koosneb sissejuhatavast osast, kus seletatakse lahti programmeerimise mõiste ning tuuakse näiteid igapäevaelust programmjuhitavate seadmete kohta. Edasi tutvustatakse programmi ja programmeerimise õppimise seisukohalt esmaseid vajalikke mõisteid ja tegevusi. I tunni lõpuks pidid õpilased olema tutvunud programmi kasutajaliidese ja *roboti* tegelaskujuga ning põhiliste käskudega, mida see täita suudab. II tunni materjal hõlmab I tunni materjali kordamist ning tutvustatakse veel programmi võimalusi, läbi mille jõutakse anda lühike ülevaade ka loogiliste avaldistest ja tingimuslausetest. II tund lõpeb kokkuvõttega ning ülevaatega erinevatest tähtsamatest programmeerimiskeeltest ning nende kasutusest. Õppematerjali kokkupanekul saadi abistavat infot programmi *RoboMind* kodulehel asuvast dokumentatsioonist⁵ (vt *RoboMind*).

Töölehed olid esitatud slaidide kujul ja seda põhjusel, et nii oli lihtne jagada edastatav info väiksemateks ülevaatlikeks ja illustratiivseteks osadeks. Teatud määral mõjutas uurija otsust ka enda programmeerimise õppimise kogemus ülikoolis. Mingil määral võib koostatud töölehti pidada ka õpiobjektideks⁶, sest need on loodud tuginedes mitmele õpiobjektidele

⁵ Tuleb märkida, et võrreldes 2012. aasta kevadega, mil töölehti koostati, võib kodulehel olev dokumentatsioon ja programmi võimalused olla muutunud.

⁶ Mõiste *õpiobjekt* kohta leiab rohkem informatsiooni HITSA Innovatsioonikeskuse poolt loodud leheküljelt <https://sisu.ut.ee/opiobjekt/1-mis-%C3%B5piobjekt>

esitatavale nõudmisele. Nimelt on töölehed juhendava iseloomuga, illustratiivsed ja koostatud selliselt, et kasutajad saaksid iseseisvalt nendega töötada. Paljude teemade ja ülesannete juures olid illustreerivad pildid oodatava tulemuse kohta. Tagasisidet õppetegevusele pakkus *RoboMindi* keskkond. Nimelt annab programm tagasisidet igale kasutajapoolsele tegevusele, mida *robotil* mingil põhjusel pole võimalik täita. Töölehed olid mõeldud arvutis kasutamiseks, kuid vastavalt võimalusele võis ka paberkandjal töölehed klassis laiali jagada.

Tundide lõpuks pidid õpilased olema saanud ülevaate programmi *RoboMind* võimalustest ja mingil määral aimu, mida tähendab programmeerimine. Need kaks tundi olid tutvustava iseloomuga ja õpilased said tutvuda selle konkreetse programmeerimise keskkonnaga. Kindlasti ei saa nende kahe tunni järel eeldada, et õppematerjaliga töötanud õpilased saavad *RoboMindis* edaspidi ise kõigega hakkama. Samuti ei saa eeldada, et nad oskavad edaspidi edukalt nii *RoboMindis* kui muudes keeltes programmeerida. Uuriija jaoks oli eelkõige oluline tekitada õpilastes huvi sellise programmeerimise kui õppimisvõimaluse vastu.

2.2 Uuritavad ja protseduur

Lisaks õppematerjalile koostati antud uurimuse käigus kaks ankeeti õpilastele ja õpetajatele programmeerimise õppimisele tagasiside saamiseks. Palve töölehtede katsetamiseks edastati 2012. aasta 2. aprillil e-posti teel rohkem kui 80 Eesti põhikooli matemaatika-, füüsika- ja informaatikaõpetajale. Vastavasisulise kaaskirja võib leida Lisas 1. Koolid valiti 2012. aasta 1. aprillis Eesti põhikoolide nimekirjast *NETI* otsingumootori rubriigis. Kuna ei olnud võimalik ennustada, kui paljud õpetajad oleksid nõus sellele palvele vastama, siis prooviti teade edastada võimalikult paljudele põhikooliõpetajatele. Seega on tegemist mugavusvalimiga. Peale selle, et õpetajad töötaksid põhikooli klassidega, ei olnud uuritavatele rohkem nõudmisi.

Kui õpetajad olid andnud nõusoleku uurimuses osalemiseks, saadeti neile e-posti teel töölehed koos juhistega. Vastavasisulise kaaskirja võib leida Lisas 2. Töölehed jõudsid uuritavateni sama 2012. aasta aprilli esimeses pooles. Kõik õpetajad peale ühe viisid tunnid läbi umbes kahe nädala jooksul peale töölehtede kätte saamist. Üks õpetaja viis tunnid läbi aprilli lõpus. Töölehtedega töötasid ja hiljem oma hinnangu nendele andsid 7 reaalinete õpetajat ja 153 õpilast.

Õpetajad, kes olid õppematerjali läbi töötanud, andsid sellest uurijale märku ning said võimaluse ankeedi teel uurijale tagasisidet anda. Vastava ankeedi võib leida Lisas 3.

Ankeedid jõudsid kõigi õpetajateni peale ühe aprilli teises pooles. Viimane õpetaja täitis ankeedi mai alguses. Õpilastele mõeldud ankeedi viide asus II tunni töölehtede lõpus, nii et ülesannete lõppedes oli võimalik kohe suunduda küsimuste juurde. Vastava ankeedi võib leida Lisas 4.

Uurimuses osales kaks meesõpetajat ja viis naisõpetajat. Kõik õpetajad peale ühe olid töötanud koolis kümme ja rohkem aastat. Vajaliku info õpetajate tööstaazi ja õpetatavate ainete kohta võib leida Tabelis 1.

Tabel 1. Õpetajate poolt õpetatavad ained ja tööstaaz

Aine(-d)	Staaž
Informaatika, Füüsika	2 aastat
Informaatika	28 aastat
Matemaatika	37 aastat
Matemaatika, Informaatika	10 aastat
Matemaatika	29 aastat
Matemaatika	12 aastat
Matemaatika	14 aastat

Neli õpetajat viisid tunnid läbi 7. klassiga, üks õpetaja 8. klassiga, üks õpetaja 6. ja 9. klassiga ning üks õpetaja 5. klassiga. Kokku katsetati töölehti kaheksas erinevas klassis. Programmeerimist õppisid 81 poissi ja 72 tüdrukut. Õpilaste arvuline jaotus klassiti on toodud Tabelis 2.

Tabel 2. Uurimuses osalenud õpilaste arv klassiti

Klass	Õpilaste arv
5	4
6	11
7	104
8	21
9	13

Kasutatud andmete uurimiseks ja töötlemiseks kasutati programmi MS Excel. Õpilaste andmete töötlemise käigus kasutati nendest ülevaate saamiseks ja kirjeldamiseks protsente

ning graafilisi vahendeid. Õpetajate vastuseid analüüsiti ja tulemustes esitati saadud kirjeldused või uurijapoolsed kokkuvõtted kirjeldustest. Ei õpetajate ega õpilaste vastuste põhjal saa teha mingeid üldistusi ega järeldusi, sest tegu pole esinduslike valimitega. Uurimuses osalenud õpetajate arv oli väga väike ning olgugi et vastanud õpilaste arv osutus lõpuks üsna suureks, siis klassiti oli osalenud õpilaste arv väga erinev. Kuivõrd oleks võimalus otsida seoseid ja võrrelda 7. klassi õpilaste vastuseid, kuid aja- ja mahupuudusel seda tööd siinkohal ette ei võetud. Samuti ei saa nelja 7. klassi puhul teha üldistavaid järeldusi kõigile 7. klassi õpilastele.

Kuna antud uurimuse käigus kasutati katseisikute abi, siis on oluline tuua välja uurimisobjektide seisukohalt neli asjaolu, mis on bakalaureusetöö kui uurimistöö eetiliste nõuete täitmisel olulised (Hirsjärvi, Remes ja Sajavaara, 2004). Võib öelda, et õpetajate seisukohalt on enamjaolt täidetud Homani (1991, viidatud Hirsjärvi jt, 2004 j) nii *informeerituse* kui ka *nõusoleku* nõuded. Õpetajaid informeeriti esmases pöördumises sellest, kes on töö autor ja mis eesmärk on uurimuse läbiviimisel. Siinkohal on oluline mainida, et uurimuse eesmärk ajas teatud määral muutus, kuid kasutati juba kogutud andmeid ega viidud näiteks läbi uut uurimust. Seega ei olnud õpetajad teadlikud uurimuse lõplikust eesmärgist. Õpetajaid teavitati esimeses pöördumises ülesannetest, mida tuleb täita, et antud uurimuses osaleda (näiteks programmi allalaadimine arvutitesse, hilisem ankeetide täitmiseks kuluv aeg jne). Samuti teavitati õpetajaid neile ja õpilastele uurija poolt tagatavast anonüümsusest. Seega võib öelda, et õpetajate puhul on täidetud *informeerituse* esimene nõue, et kõik olulisem uurimuse läbiviimise seisukohast oleks ka katseisikutele teada. Autori hinnangul oli kirjade teel antud informatsioon ülevaatlik ja õpetajatele arusaadav, sest informatsiooni puuduse või kahetimõistmise suhtes ei saabunud uurijale ühtki pretensiooni. Seega võib öelda, et õpetajate puhul on täidetud teine *informeerituse* nõue, et katseisikud on mõistnud uurimuse läbiviimisi olulisemaid aspekte. Õpilaste puhul ei saa aga rääkida nende nõuete täitmisest. Seda seetõttu, et otsuse õppematerjali katsetada tegi õpetaja. Uurijale pole teada, mil määral õpilased said otsustusprotsessis osaleda ning kuidas neile selgitati õppeprotsessi käiku. Seda ei uuritud töö käigus. Ise küsimus on, kas õpilased oleksid mõistnud või osanud hinnata pakutavat informatsiooni. Samuti tuleb märkida, et uurija ei pöördunud lapsevanemate poole küsimaks luba õpilaste osalemiseks antud uurimuses. Uurijale pole ka teada, mil määral õpetajad ise lapsevanemaid õppematerjali katsetamisest teavitasid.

Nõusoleku nõuete kohaselt peavad katseisikud olema pädevad andma adekvaatseid hinnanguid ja tegema seda vabatahtlikkuse alusel. Õpilaste puhul ei saa rääkida vabatahtlikkuse nõude täitmisest, sest nagu öeldud, pole uurijale teada, mil määral arvestati

õpilaste soovi antud uurimuses osaleda. Võib öelda, et õpetajad olid oma otsustes vabad. Kõigile õpetajatele edastati viisakas palve ühe e-kirja teel. Pöördumised õpetajate poole võib leida Lisades 1 ja 2. Ainus nõue valimile oli, et osalejateks oleksid põhikoolide õpilased ja õpetajad. Kuna antud uurimuse käigus uuriti õpilaste ja õpetajate huvi programmeerimise vastu ning uurimuses osalemine ei nõudnud eelnevaid teadmisi, siis võib öelda, et katseisikud olid piisavalt pädevad andmaks adekvaatseid hinnanguid.

2.3 Mõõtevahendid

Õpetajatelt ja õpilastelt hinnangute saamiseks õppematerjalile kasutati kahte ankeetküsitlust. Mõõtevahendiks valiti elektrooniline ankeet kahel põhjusel. Esiteks seeläbi taheti õpetajatele ja õpilastele tagada anonüümsus töölehtede katsetamisel. Teiseks paiknesid tundide läbiviimisega nõustunud õpetajad ja õpilased üle Eesti, seega oleks keeruline ja aeganõudev olnud teistsugust mõõtevahendit, näiteks intervjuud õpetajate puhul, kasutada. Ankeedid koostati autori enda poolt. Valiidsuse tagamiseks andis nendele oma hinnangu kaasjuhendaja. Õpilaste ankeedi sisereleiaablust mõõdeti SPSS Statistics abil Cronbachi alfat kasutades ja vastav näitaja tuli alla 0,7. Õpetajatele ja õpilastele saadetud ankeedid võib vastavalt leida Lisades 3 ja 4.

Õpetajate ja õpilaste ankeetides oli vastavalt 21 ning 16 vaba- ja valikvastustega küsimust. Õpetajatel paluti anda põhjalikum hinnang programmeerimise õpetamise võimalustele ja vajalikkusele põhikoolis. Õpetajad kirjeldasid tundide läbiviimist vabas vormis ning andsid hinnangu valitud programmile ja töölehtedele. Nii õpetajate kui õpilaste puhul uuriti nende varasema programmeerimise alase kogemuse olemasolu. Õpilased andsid samuti hinnangu programmile, töölehtedele ja programmeerimise õpetamisele. Võib öelda, et õpetajate ankeet sisaldas rohkem avatud ja õpilaste ankeet rohkem valikvastustega küsimusi.

III TULEMUSED

3.1 Tulemused õpetajate tagasiside põhjal

3.1.1 Õpetajate eelnev programmeerimise õpetamise ja õppimise kogemus.

Kuna programmeerimise õpetamine ei ole Eestis põhihariduse seisukohalt kohustuslik, siis oli põhjust uurida kui paljud uurimuses osalevatest õpetajatest kas huviringides või koolitundides seda teemat käsitlenud on. Selgus, et seitsmest õpetajast kaks on gümnaasiumi tasemel programmeerimist varem õpetanud. Esimene neist oli informaatikaõpetaja, kes oli ainult ise internetis teadmisi selle kohta omandanud ning gümnaasiumiõpilastele huviringis *Basic ja Visual Basic* programmeerimiskeeli õpetanud. Teine oli matemaatika- ja informaatikaõpetaja, kes oli Tartu Ülikoolis informaatikaõpetajate kutsekursustel programmeerimist õppinud ja 12. klassis sellistes keeltes nagu *Pascal*, *JavaScript* ja *HTML* programmeerimist õpetanud. Mõlemal õpetajal oli koolis õpetamise staaži vähemalt 10 aastat. Ainult ülikoolis olid programmeerimist õppinud viis õpetajat.

Töö autor püstitas 4 uurimisküsimust, millele sooviti õpetajate tagasiside põhjal vastused leida. Nendest lähtuvalt kirjeldatakse siin edaspidi saadud tulemusi.

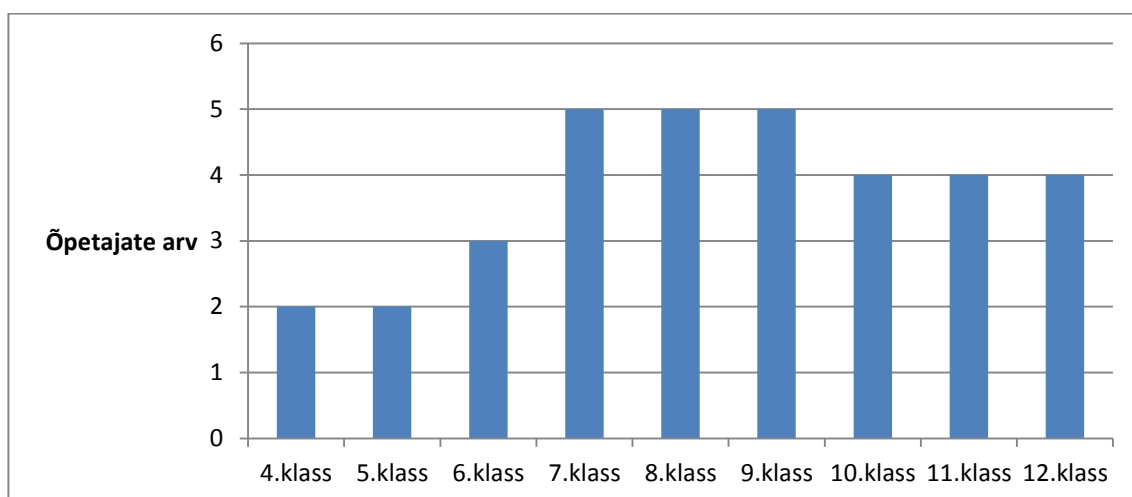
Küsimused olid järgmised:

1. Kuidas suhtuvad õpetajad programmeerimise õpetamisesse põhikoolis?
2. Kuidas võiks programmeerimist õpetajate arvates põhikooli tasemel õpetada?
3. Kuidas hindavad õpetajad koostatud töölehtede sobivust programmeerimise õpetamiseks?
4. Kuidas hindavad õpetajad valitud programmi *RoboMind* sobivust programmeerimise õpetamiseks?

3.1.2 Õpetajate seisukohad programmeerimise õpetamise suhtes üldhariduskoolides. Kolm õpetajat, kellel puudus varasem programmeerimise õpetamise kogemus, pidasid võimalikuks koolis programmeerimise õpetamist. Põhjendati oma valikut mõtlemisvõime arendamisega ning õpilaste aktiivsuse ja huviga. Üks õpetaja tõi oma põhjenduses siiski välja, et sellel ei olnud seost 7. klassi matemaatikaga. Seega selle õpetaja puhul ei olnud päris selge, miks ta valis vastusevariandi *võimalik jah*. Üks nendest õpetajatest,

kes olid ka varem koolis programmeerimist õpetanud, tõi välja, et õpetaks RoboMindi põhikooli klassidele. Teine õpetaja leidis, et põhikoolis saaks programmeerimist õpetada kiirematele õpilastele lisateemana. Kaks õpetajat leidsid, et ajapuuduse tõttu nad koolis seda teemat ei puudutaks. Üks neist lisas ka, et ei tunne ennast piisavalt pädevana selles valdkonnas. Samas tõi üks neist õpetajatest välja: „*Kui muudetakse ainekava, siis oleksin kiüll nõus, sest abikaasa tegeleb programmeerimisega ja saaksin ise ka areneda*“ ja lisab veel „*hetkel ei kujuta hästi ette, sest põhikooli eksamil tuleb ülesandeid siiski paberile lahendada ja ilma programmide abita.*“ Teine õpetaja lisas seejuures: „*Kui matemaatikatunde oleks rohkem, siis võiks paar korda kuus programmeerida.*“ Kõik kolm õpetajat, kes mainisid tagasisides, et ajapuuduse tõttu nad kohustuslikus korras programmeerimise õpetamisele pühenduda ei jõuaks, olid matemaatikaõpetajad. Tuleb märkida, et ükski õpetaja ei valinud vastusevarianti, et tahaks kindlasti edaspidi programmeerimist koolis õpetada.

Joonis 5 kirjeldab õpetajate hinnanguid, millistele klassidele sobiks programmeerimist õpetada. Tuleb märkida, et ühe õpetaja vastust ei arvestatud tabelis, sest ta ei toonud välja, mis klassides võiks programmeerimist õpetada, vaid rõhutas ainult, et see peaks toimuma lisaaines. Üks õpetaja tegi otsuse lähtuvalt sellest, mis klassist alates hakkavad õpilased seal koolis arvutiõpetust õppima. Mitu õpetajat tõi välja ainult selle, mis klassist alates võiks programmeerimist õpetada. Kaks õpetajat tõi selged vahemikud. Seega tabel koostati eeldusel, et kui nõ ülemist piiri ei antud, siis võis sellena lugeda üldhariduskooli viimast klassi, mille seadiski piiriks üks õpetaja.



Joonis 5. Õpetajate valikud, millistele klassidele nad programmeerimist õpetaksid

Eraldi uuriti, kas just põhikoolis peaks programmeerimist õpetama ning selgus, et neli õpetajat leiavad, et see oleks vajalik. Neist kaks õpetajat tõid välja, et sel juhul peaks programmeerimise ülesanded olema õpilastele huvipakkuvad. Autor toob välja siinkohal ühe õpetaja vastuse: „*Põhikoolis võiksidki olla sellised robomind tüüpi mänguülesanded.*“ Kaks õpetajat tõid aga välja, et seda võiks teha valikaine või ringitunni raames. Üks õpetaja ei vastanud sellele küsimusele täpselt, kuid lisas, et programmeerimise õpetamine võiks olla siiski reaalkallakuga keskkoolide ja ülikoolide pädevuses. Kolm õpetajat täpsustasid sellele küsimusele vastas, et programmeerimist peaks õpetama just informaatikatunnis. Üks matemaatikaõpetaja tõi välja, et kuna eksamil ei nõuta selliseid oskusi, siis ta ei näe programmeerimist matemaatika ja informaatika lõimimise võimalusena.

Õpetajatelt uuriti, millistes õppeainetes programmeerimist võiks õpetada. Tabelis 3 on kujutatud vastusevariandid ja info selle kohta, mitu õpetajat iga vastusevarianti valis.

Tabel 3. Õppeained, kus õpetajad õpetaksid programmeerimist

Aine	Õpetajate arv
Informaatika	7
Matemaatika	5
Füüsika	2
Muu aine	1

Võib öelda, et kõik informaatika- ja füüsikaõpetajad leidsid, et programmeerimist võiks koolis õpetada ja sellega võiks nende tunnis tegeleda. Samuti tuleb märkida, et kõik informaatikaga seotud õpetajad on väljaspool ülikooli programmeerimises teadmisi omandanud. Küsimus oli nii püstitatud, et need õpetajad, kes ei arva, et programmeerimist peaks koolis õpetama, pidid siinkohal samuti oma valiku tegema. Õpetajatel paluti põhjendada, miks peaks põhikoolis programmeerimist õpetama või tuua välja vastupidine seisukoht. Kõik õpetajad tõid välja põhjused, miks oleks kasulik põhikoolis programmeerimisega tegeleda. Neli õpetajat tõid ühe põhjusena välja loogilise mõtlemise. Kaks õpetajat leidsid, et see võiks tekitada huvi IT-valdkonna vastu. Lisaks sellele toodi põhjustena välja ruumilise, järjestatud, ja üldiselt mõtlemise arendamine, arvutite mõistmine ja seos igapäevaeluga. Õpetajatel paluti ka eraldi välja tuua konkreetsed oskused ja teadmised, mida programmeerimine võiks arendada. Lisaks eelpool mainitule leiti, et programmeerimine arendab üldistamis-, planeerimis-, seostamis- ja funktsionaalset lugemisoskust, süsteemset mõtlemist ja oskust ülesannet tükkideks lahata.

3.1.3 Õpetajate hinnangud töölehtedele ja programmile. Neli õpetajat seitsmest jäid koostatud töölehtedega rahule. Nendest õpetajatest, kes töölehtedele mingil määral kriitilisemat tagasisidet andsid, esimene tõi välja, et kasutas ainult I tunni materjali ja tegi II tunni materjali sama programmi abil ise. Teine õpetaja leidis, et ülesandeid oleks võinud rohkem olla, sest palju võimalusi *RoboMindi* keskkonnas jäi tutvustamata. Kolmas õpetaja leidis, et nõ katsetamise ülesannete asemel oleks võinud konkreetsemad ülesanded olla. Ühel töölehti kritiseerinud õpetajatest oli eelnev programmeerimise õpetamise kogemus. Kaks õpetajat, kes töölehtedele positiivset vastukaja andsid, tõi välja, et töölehed olid lihtsad ja arusaadavad. Neist õpetajatest ühele tundus, et mõnede 9. klassi õpilaste jaoks võisid võrreldes 6. klassiga ülesanded isegi liiga lihtsad olla.

Kolm õpetajat seitsmest hindasid programmi *RoboMind* sobivaks programmeerimise õpetamiseks. Üks neist õpetajatest leidis, et õpetaks *RoboMindi* abil programmeerimist põhikooli klassides. Antud õpetaja katsetas töölehti 5. klassi õpilastega. Programmi juures toodi positiivsena välja järgmised omadused: lõbus keskkond, kohene tagasiside ja hea vaheldus matemaatikale. Ülejäänud õpetajatest üks leidis, et programm oleks võinud *vabam* olla, kuid andis kokkuvõttes positiivse hinnangu. Õpetaja võis mõelda selle all, et programmeerimise võimalusi kitsendasid teatud asjaolud, kuid midagi täpsemat ankeedi vastustest ei selgunud. Teine õpetaja ei hinnanudki programmi enda seiskohast, vaid tõi välja, et tema õpilastele meeldiks rohkem programmi *Scratch* abil programmeerida. Tegu oli 7. klassi õpetajaga. Kolmas õpetaja ei hinnanud samuti programmi. Viimane hinnangu andnud õpetajatest tõi aga välja, et õpilastele valmistas programmeerimise osa⁷ programmi juures siiski raskusi. Samuti tõi see õpetaja välja, et osad õpilased ei saanud juhendi täitmisega hakkama, mis võib olla suunatud kriitika töölehtedele. Antud õpetaja katsetas töölehti 6. ja 9 klassis, kuid ankeedi vastustest ei selgunud, kas mõlemas klassis tekkis see probleem.

3.2 Tulemused õpilaste tagasiside põhjal

3.2.1 Õpilaste eelnevad teadmised programmeerimise kohta. Tuleb märkida, et peaaegu kõik (96%) selles uurimuses osalenud õpilased olid mingil määral informaatikat õppinud. Õpilastelt uuriti, kas ja kust nad on kuulnud midagi programmeerimise kohta enne antud uurimuse käigus läbiviidud tunde. Selgus, et 57% õpilastest olid kas kuulnud selle IT-valdkonna osa kohta midagi või on ise programmeerimisega kokku puutunud. Tuleb märkida, et osa nendest õpilastest (26%) täpsustasid ebamääraselt seda, kust nad on kuulnud midagi

⁷ Nimelt on programmis võimalik tegelaskuju *robotit* juhtida nii puldi kui *programmijuppide* koostamise abil.

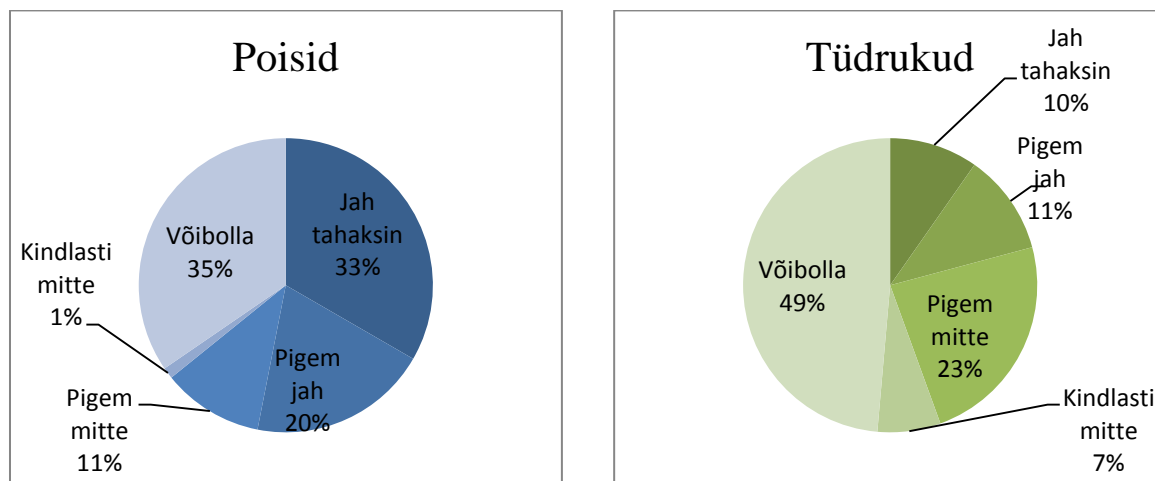
programmeerimise kohta. Seega järgnevalt toodud arvulised näitajad võivad olla ebatäpsed. Nendest õpilastest 22% on programmeerimisega koolitunnis kokku puutunud ja 9% on huviringis programmeerimisega tegelema. Veel toodi infoallikana välja internet, televisioon, sõbrad ja tuttavad, pere ja mängud. Tuleb märkida, et kaheksa õpilast, kellest seitse olid poisid, on arvutis mängides programmeerimisega tegelema. Leidsid õpilasi, kes tõid välja, et on ise programmeerimise kohta rohkem teadmisi püüdnud omandada. Kõik õpilased, kes ise mingil määral programmeerinud olid, käisid vähemalt 7. klassis. Kindlasti sõltub õpilaste endi kogemus sellest, mida ja kui kaua neil lubatakse kodus arvutis teha. Võib oletada, et nooremate kui 7. klassi õpilaste arvutikasutust piiratakse rohkem. Samas võivad programmeerimist nõudvad mängud olla katsetamiseks ja uudistamiseks noorematele kui 7. klassi õpilastele veel liiga keeruliseks väljakutseks.

Töö autor püstitas 3 uurimisküsimust, millele sooviti õpilaste antud tagasisidest lähtuvalt vastused saada. Nendest lähtuvalt kirjeldatakse edaspidi ka tulemusi õpilaste kohta.

Küsimused olid järgmised:

1. Kuidas suhtuvad õpilased programmeerimise võimalikku õppimise põhikoolis?
2. Kuidas hindavad õpilased koostatud töölehtede huvitavust ja keerulisust?
3. Kuidas hindavad õpilased valitud programmi RoboMind?

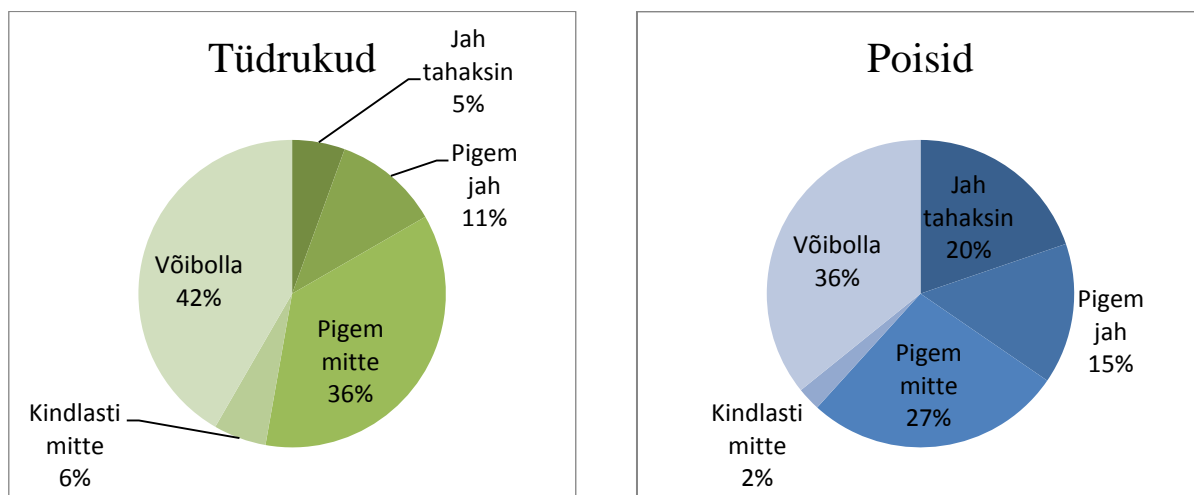
3.1.2 Õpilaste suhtumine programmeerimise õppimisse. Neid õpilasi, kes tundsid koolis programmeerimise õppimise vastu huvi, oli kokkuvõttes rohkem (38%) kui neid, kes olid pigem või täiesti selle õppimise vastu (21%). Kõige rohkem oli aga neid, kes ei osanud konkreetset seisukohta võtta (41%). Nende seas leidsid selliseid õpilasi, kes tundsid siiski huvi programmeerimise vastu ja ka neid, kellele selline tegevus siiski huvi ei paku. Joonisel 9 on toodud selles küsimuses õpilaste arvamuste jagunemine soolises lõikes.



Joonis 9. Õpilaste soov programmeerimist õppida soolises lõikes

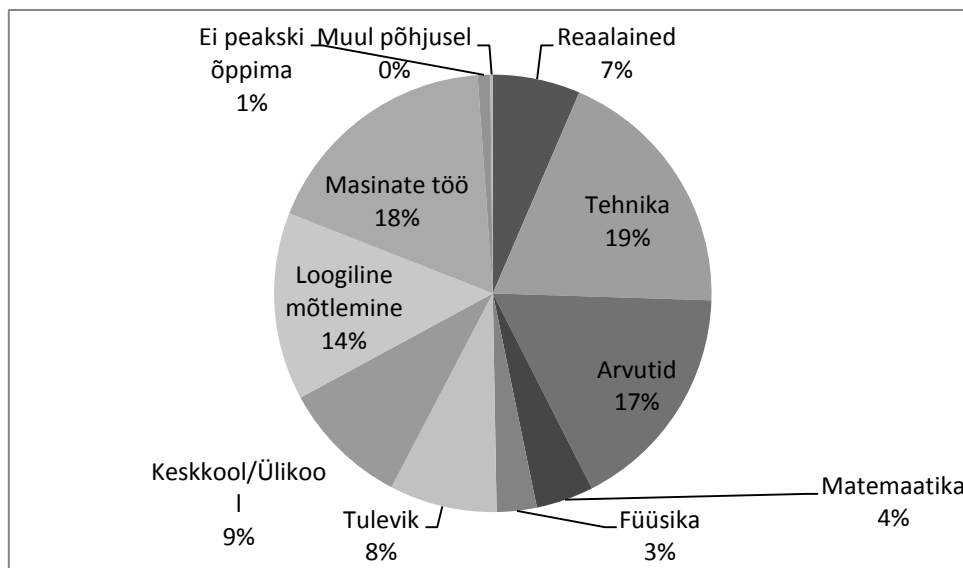
Nende õpilaste seas, kes tahaksid programmeerimist koolis kindlasti või pigem õppida, toodi enim põhjendustena välja, et see tegevus on huvipakkuv ja meeldiv. Leidus ka õpilasi, kes viitasid, et tahaksid sellel erialal tööd saada või et see oleks tuleviku mõttes kasulik, kuid neid oli siiski väga vähe. Nende seas, kes pigem mitte või üldse ei tahaks koolis programmeerimisega tegeleda, toodi enim põhjendusena välja, et see tegevus on ebahuvitav, keeruline või igav.

Kuna robotikaringides tegeletakse samuti programmeerimisega, siis oli põhjust uurida õpilastelt, kas nad on kuulnud midagi robotikast ning kuivõrd huvitatud oleksid nad selle õppimisest koolis. Leidus 61% neid, kes sellest midagi kuulnud olid ja 16% neid, kes polnud. Ülejäänud õpilased ei osanud sellele küsimusele vastata. Joonis 11 annab täpsema ülevaate õpilaste suhtumise kohta robotika võimalikku õppimisse soolises lõikes.



Joonis 11. Õpilaste robotika õppimise soov soolises lõikes

Õpilastel paluti valida põhjuseid, miks peaks koolis programmeerimist õpetama. Esimeses osas variantidest pakuti välja erinevaid valdkondi, mille vastu programmeerimine võiks huvi äratada (näiteks tehnika või füüsika). Teises osas variantidest pakuti välja, kus võiks programmeerimist edaspidi vaja minna (näiteks keskkoolis) või aitaks selle teema õppimine kaasa loogilisele mõtlemisele ja masinate tööpõhimõtetest arusaamisele. Samuti oli õpilastel võimalus valida, et programmeerimist ei peakski õppima või tuleks seda teha muul põhjusel. Joonis 12 annab täpsema ülevaate õpilaste valikute kohta soolises lõikes.



Joonis 12. Õpilaste poolt valitud põhjused, miks peaks programmeerimist koolis õppima.

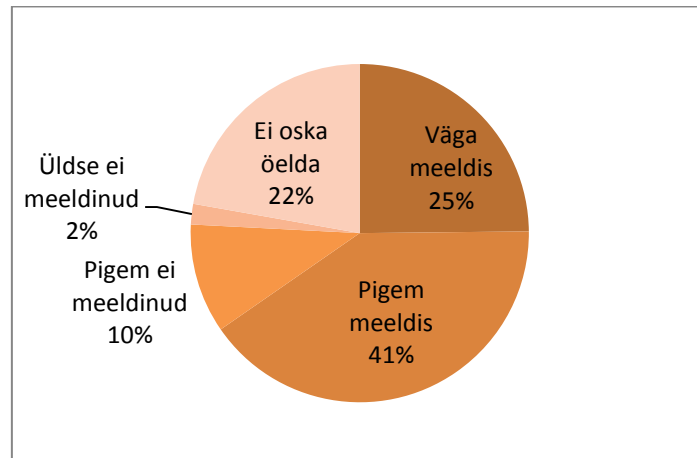
3.2.3 Õpilaste hinnangud töölehtedele ja programmile. Õpilastel paluti hinnata hinnata tundide huvitavust. Vastustest selgus, et 61% õpilastest pidasid läbiviidud tunde huvitavaks ja 31% ei osanud hinnata tunde. Poistest 65% ja tüdrukutest 56% leidsid, et tunnid olid huvitavad. Nendest õpilastest, kes hindasid tunde huvitavaks, 10% pigem või üldse ei tahaks programmeerimist koolis õppida ja 30% olid kahtleval seisukohal.

Õpilastel paluti hinnata läbiviidud tundide keerukust. Samuti said õpilased siinkohal märkida, kui neil hakkas tunnis igav ning täpsustada mitut vastusevarianti valides, mis igavust võis põhjustada. Võib öelda, et 48% õpilastest leidsid, et tunnid olid lihtsad nende jaoks ja 33% leidsid, et tunnid olid piisava raskusastmega. Üle 50% poistest ja 42% tüdrukutest leidis, et tunnid olid lihtsad. Nii poiste kui tüdrukute seas oli 30-35% õpilasi, kes leidsid, et tunnid olid piisavalt rasked. Tabel 4 annab täpsema ülevaate õpilaste hinnangute kohta.

Tabel 4. Õpilaste hinnangud tundide keerukusele

Hinnang keerukusele	Õpilaste arv
Lihtne oli	73
Piisavalt raske oli	50
Igav oli	16
Lihtne ja piisavalt raske oli	6
Igav ja lihtne oli	3
Liiga raske oli	3
Liiga raske ja igav oli	1
Piisavalt raske ja igav oli	1

Õpilastel paluti hinnata programmi *RoboMind*. Joonisel 12 on välja toodud, kuidas jagunesid õpilaste valikud protsentuaalselt.



Joonis 12. Õpilaste hinnangud programmile

Tuleb märkida, et nende õpilaste seas, kellele valitud programm väga meeldis, ei olnud keegi pigem või täiesti programmeerimise õppimise vastu. Õpilastel paluti oma hinnanguid programmile põhjendada. Programmi kritiseerijaid leidis ka nende hulgas, kes valisid, et see programm mingil määral neile meeldis. Mitmel õpilasel oli soov veel rohkem või raskemaid ülesandeid lahendada nii *RoboMindis* kui üldisemalt programmeerides. Negatiivsena toodi välja, et programm oli igav, keeruline, ebahuvitav või liiga lihtne. Positiivsena toodi välja, et programmis tegutsemine oli huvitav, lõbus, lihtne ja pakkus uusi teadmisi. Leidis ka mitu õpilast, kes ei osanud oma vastust põhjendada.

IV ARUTELU

Antud uurimuse läbiviimisel loodeti äratada õpetajate ja õpilaste huvi programmeerimise õppimise võimaluse vastu. Võib öelda, et õpetajate huvi programmeerimise õpetamise suhtes oli pigem tagasihoidlik. Enamus õpetajaid leidsid, et programmeerimise õpetamine oleks vajalik, kuid pidasid selle teema õpetamist tulevikus võimalikuks, aga mitte kindlaks ettevõtmiseks. Enamus vastanud õpilastest olid samuti ebakindlad koolis programmeerimise õppimise soovis. Suurem osa õpilastest leidis siiski, et läbiviidud tunnid olid huvipakkuvad. Kuna ankeedis polnud täpsustatud, mis tundides võiks selline võimalus tekkida, siis võib oletada, et õpilastele tunduvad õppeained üsna mahukad ja seetõttu kardetakse igasuguse uue ettevõtmise puhul tundides koormuse tõusu. Tuleb märkida, et töölehed olid koostatud nii, et õpilased saaksid võimaluse korral ka iseseisvalt ja ilma õpetaja abita tegutseda. Samuti tuleb märkida, et selle teemaga tegeleti väga lühikest aega. Näiteks McInerney (2010) toob välja, et Scratchiga programmeerimise õppimiseks töötati erinevate klassidega 5-32 nädala jooksul. Õppematerjal loodi eesmärgiga tutvustada lihtsal kujul mängulist programmeerimist ja programmi *RoboMind* võimalusi nii õpetajatele kui õpilastele. Samas võib õpetajate ja õpilaste ebakindluse põhjal oletada, et töölehtede koostamine nii väheeks ajaks või sellisel kujul ei aidanud piisavalt kaasa koolis programmeerimise soovile õpilaste ja õpetajate seas.

Ankeetide abil taheti teada saada, mil viisil õpetajate arvates sobiks programmeerimist koolis õpetada ning mis põhjusel seda peaks tegema. Kõik õpetajad ja valdav enamus õpilasi leidsid, et programmeerimise õpetamisel on erineval viisil kasulik mõju. Suur osa õpetajaid ja õpilasi leidis, et loogilise mõtlemise arendamine on üks põhjusi, miks peaks programmeerimist õppima. Võib öelda, et see on kindlasti üks oluline väljund programmeerimise õppimisel. Kelleher (2009) leiab, et positiivne kogemus programmeerimises tuleks kasuks paljudel erialadel ja aitaks avada informaatika eriala suuremale hulgale õpilastele. Osa õpetajaid ja suur osa õpilasi nägid koolis programmeerimise õppimises kasulikku otstarvet õpilaste tulevikus tehtavatele valikutele. On huvitav tõdeda, et väga suur osa uuritavatest leidsid, et programmeerimise õppimine on õpilastele erineval viisil kasulik, kuid tööst ei kajastunud suuremat entusiasmi seda tulevikus õpetada või õppida.

Tuleb märkida, et informaatikaõpetajad nägid programmeerimise õpetamiseks rohkem võimalust kui matemaatikaõpetajad. Kuna põhikoolilõpetajad peavad sooritama matemaatikaeksami, siis võib oletada, et matemaatikaõpetajad ei näe võimalust, et ettenähtud õppematerjalidega töötamise kõrvalt tegeleda põhjalikumalt programmeerimise ülesannetega. Samas kinnitavad mõned autorid, et ka mitte reaalinete õpetajad (Wolz jt, 2011) või koolieelse lasteasutuse õpetajad (Fesakis & Serafeim, 2009) saavad näiteks Scratchiga programmeerimisega väga hästi hakkama, kui neile selleks neile võimalust anda ning võivad kaaluda ka edaspidi selliste õppimisvahendite kasutamist koolis.

Õpetajate antud vastustest kajastus, et põhikoolis 7.-9. klassis sobiks enim programmeerimist õpetada. Mitmed autorid on uurinud edukalt võimalusi just põhikooli õpilastele programmeerimist õpetada. Gweon, Ngai ja Rangos (2005) toovad välja, et suutsid äratada põhikooli tüdrukute huvi programmeerimise vastu. Repenning (2012) kinnitab, et õpilased peaksid juba põhikooliklassides tutvuma programmeerimisega. Brauner, Leonhardt, Ziefle ja Schroeder (2010) leidsid aga, et juba 7. klassi õpilaste tasemel on tüdrukud vähem enesekindlad üldiselt tehnilistele probleemidele lahendusi otsides. Selleks et tõsta huvi teaduse ja tehnika vastu, tuleks nende autorite meelest alustada erinevate huvitavate abipakkuvate tegevuste tutvustamist juba varem kui 7. klassi tasemel, et vältida teatud hoiakutes suuremaid soolisi erinevusi. Antud töös olid enamus programmeerimise kogemusega õpilastest poisid, kes olid ise kodus vastavaid oskusi nõudvaid mängu mänginud. Antud töö autor ei oska öelda, kas leidub uurimus, mis kinnitavad sellist tendentsi. Samuti tuleb märkida, et poisse, kes tahaksid kindlasti ka tulevikus programmeerimisega koolis tegelda, oli kolm korda rohkem kui sellisel seisukohal olevaid tüdrukuid. Brauner, Sullivan ja Bers (2013) nendivad, et hetkel on väga vähe uuritud soolisi erinevusi noorte õpilaste puhul, sest programmeerimise õppimine ise on nii noore huvigrupi puhul veel uus teema.

Kolm õpetajat tõid välja, et programmeerimist võiks õpetada koolitundides ja mitte koolivälisel ajal. Repenning (2012) leiab, et programmeerimise õppimise võimaluse pakkumisest pärast kooli huviringide näol ei piisa ning leiab üldiselt, et põhikoolides peaks informaatikat õpetama süsteemselt ja kohustuslikus korras. Siiski võib öelda, et Eestis ei ole võetud sellist suunda, et üle Eesti muutuks informaatika teatud klassidele kohustuslikuks õppeaineks. Samuti ei saa öelda, et Eestis võiksid lähitulevikus hakata põhikoolide õpilased programmeerimist õppima, sest arvatavasti pole selleks piisavalt koolitatud õpetajaid ega ka haridusüldsuse poolt üldtunnustatud vahendeid.

Kuna antud uurimuses osalenud õpetajate arv oli väike ja õpilaste arv oli klassiti väga erinev, siis ei saa antud töö tulemuste põhjal midagi järeldada ega üldistada. Samuti võib

öelda, et mõned õpetajad ei jäänud koostatud õppematerjaliga rahule ja ka programmi RoboMind suhtusid enamuse õpetajaid mõningase kriitilisusega. Kriitika ja ebakindluse nii õpilaste kui õpetajate seas võisid tingida niivõrd uudse teema kohta väikesemahuline õppematerjal ja lühiajaline katsetamine. Enamus poisse ja väga suur osa tüdrukuid leidsid, et tunnid olid lihtsad. Samuti oli märkimisväärne osa õpilasi, kellel hakkas mingil põhjusel tunnis igav. Uurimuse tulemustest ei peegeldunud mingisugust nähtavat seost töölehtede raskusastme ja klassitaseme vahel, seega ei saa ka oletada, millistes klassides sobib rohkem loodud töölehti katsetada. Antud töö autor leiab, et töölehtedel olevate ülesannete valik oleks võinud olla suurem. Siiski peab märkima, et tasuta kasutamiseks on programmi võimalused piiratud ja võib öelda, et kahe koolitunni jooksul ei ole algajatega võimalik ka kõiki olemasolevaid võimalusi läbi proovida. Seega on mõlemat ülesannete komplekti mingil määral täiendatud lisaülesannetega.

Samuti võib uurimuse ühe piiranguna välja tuua ankeedid. Järgnevalt antakse mõned soovitusel ankeetide muutmiseks ja seda mitmel põhjusel. Esiteks on mõningased muutused vajalikud, sest algselt oli töö eesmärk uurida erinevate reaalinete lõimimise võimalusi, kuid töö eesmärk ajas muutus ja hiljem keskenduti teoreetilises osas rohkem informaatikaõpetuse sisule. Seega olid mõned küsimused otstarbetud. Teiseks tingis suur hulk avatud küsimusi õpetajate ankeedis olukorra, kus uurija oleks soovinud saada mõnede vastuste puhul täiendavat informatsiooni, kuid valitud andmekogumismeetod ei võimaldanud seda. Autori arvates aitaks ankeetidesse viidavad muudatused parandada nende kasutamisevõimalusi ja arendada neid efektiivsemaks. Seetõttu antakse siinkohal soovitusel mõnede küsimuste muutmiseks.

Esmalt tuleb juttu õpetajate ankeetides vajalikest muudatustest. Edasise uurimise seisukohalt oleks mõistlik esmalt keskenduda just informaatikaõpetajate seisukohtadele ja hinnangutele. Loomulikult võiksid uuritavateks olla ka mitme aine õpetajad, kes sealhulgas õpetavad koolis ka informaatikat. Seega tuleks õpetajate ankeedist eemaldada 14. ja 16. küsimus ja keskenduda 20. küsimuses informaatikaõpetajate kogemusele. Küsimustele 11 ja 13 võiks pakkuda edaspidi valikuvariante. Küsimuse 11 valikuvariandid võiksid sisaldada vastusevariante *koolitus, ülikoolis läbitud kursus* ja *iseseisev uurimine internetis*. Siinkohal võiks paluda ka täpsustust valitud vastusevariantidele. Sõnastused ei ole siinkohal täpsed. Küsimuse 6 võiks muuta mitmeks küsimuseks, kus uuritakse eraldi, *milliste klassidega, kui tihti* ja *kas koolitunnis või kooliväliselt* on programmeerimisega tegeletud. Siinkohal võiks paluda õpetajatel põhjendada oma valikuid, miks just sellisel viisil on programmeerimist õpetatud. Küsimustele 8 ja 9 võiks lisada juurde täpsustava küsimuse, mille abil uurida, kas

õpetajad oleksid altimad programmeerimist koolis õpetama, kui neile pakutaks vastavat koolitust. Siinjuures oleks ka huvitav teada kuivõrd õpetajad on teadlikud hetkel Eestis pakutavatest koolitamise võimalustest. Kuna osad õpetajad olid vastates üsna napisõnalised, siis võiks õpetajate ankeedi 18. ja 19. küsimuse jagada kaheks ja paluda vastajatel tuua nii töölehtede kui programmi puhul välja nii positiivset kui negatiivset.

Õpilaste ankeetides oleksid vajalikud järgmised muudatused. Edaspidi oleks mõistlik katsetada töölehti samade klassidega, näiteks ainult 7. klassidega või kindlustada, et kõikidest klassidest võtaks osa mingile vahemikule vastavas arvus õpilasi. Antud uurimuses osales näiteks liiga vähe 5. klasside õpilasi. Küsimused 4 ja 5 võiks asendada ühe küsimusega, mille abil uurida õpilaste varasemaid programmeerimisalaseid teadmisi. Küsimus võiks olla vastusevariantidega, mis sisaldaks sealhulgas *ringitundi*, *koolitundi* ja *iseseisvat õppimist internetis*. Küsimus 10 sisaldab vastusevarianti *igavus*, mis ei ole selle küsimuse seisukohalt oluline. See vastusevariant tuleks sealt eemaldada. Küsimuse 11 vastusevariantide arvu tuleks vähendada. Kuna edaspidi võiksid uuritavateks olla informaatikaõpetajad, siis oleks mõistlik õpilastelt uurida, kas ja mil määral neile meeldiks õppida programmeerimist just informaatikatunnis.

Mitmed autorid on kirjeldanud edulugusid õpilaste ja õpetajate kokkupuutest programmeerimise õppimisega. Repenning (2012) toob näite õpetajast, kelle tunnis varem õpiti trükkimist ja PowerPointi kasutama, kuid seejärel hakati programmeerimisega tegelema ning õpilaste seas leidis see positiivset vastukaja. Siegle (2009) toob välja, et on ise programmeerimist edukalt õpetanud 4. klassi õpilastele. Fesakis ja Serafeim (2009) toovad positiivse näite õpetajakoolituse üliõpilastest, kes õppisid Scratchiga programmeerimist ja kellel valdavalt puudus eelnev programmeerimise õppimise kogemus. Autorid toovad välja, et nende üliõpilaste arv, kes tahaksid ka tulevikus programmeerimist hariduslikul eesmärgil kasutada, peale vastava kursuse läbimist suurenes. Kuna antud uurimuses osalenud kõik informaatikaõpetajad olid huvitatud programmeerimise õpetamisest koolis oma tundides ja rohkem leidis õpilasi, kes sellest huvitatud olid, kui neid, kes ei olnud, siis võiks edaspidi uurida programmeerimisega tegelemise võimalusi põhikooli tasemel just informaatikatundides.

Kokkuvõte

Töö pealkiri: RoboMind kui võimalus äratada põhikooli õpetajate ja õpilaste huvi programmeerimise vastu

Võib oletada, et suur osa Eesti põhikoolide õpilasi ei tea, mida kujutab endast programmeerimise õppimine, olgugi et lastele mõeldud programmeerimiskeskondade valik on ajas suurnenud ja täiustunud. Nii Eestis kui välismaal tegutsevad haridusedendajad järjest aktiivsemalt selles suunas, et programmeerimise õppimise võimalus jõuaks rohkema ka lasteni ja seda eelkõige läbi kooliinformaatika. Programmeerimise õppimine on mitmel tasemel õpilaste mõtlemisoskusele kasulik. Samuti on võimalik läbi selle tegevuse tunda rõõmu millegi uue loomisest, kujundamisest ja teistega jagamisest ning saada elementaarsed informaatikaalased teadmised, mida võib nii edasi õppides kui töötades vaja minna.

Käesoleva bakalaureusetöö eesmärgiks oli koostada põhikooli õpilastele programmeerimise lühiajaliseks õppimiseks õppematerjal ning saada sellele ja programmeerimise õppimise võimalusele üldisemalt õpetajatelt ja õpilastelt tagasisidet. Õppematerjal koosneb kahest töölehtede komplektist ja on mõeldud hinnanguliselt kaheks koolitunniks. Õppematerjaliga töötasid 7 reaalainete õpetajat ja nende õpilased. Enamuse uurimuses osalenud õpetajate ja õpilaste arvates on programmeerimise õppimisel erineval viisil kasulik mõju õpilaste mõtlemisoskusele ja tulevikule. Valdav enamus õpilasi hindasid tunde huvitavaks ja lihtsaks ning hindasid valitud programmi *RoboMind* üldiselt meeldivaks. Siiski jäi nii õpetajate kui õpilaste puhul soov kindlasti programmeerimisega koolis ka edaspidi tegeleda tagasihoidlikuks. Lähtuvalt osade õpetajate hinnangutest vajasisid töölehed teatud määral täiendamist, mida ka tehti.

Märksõnad: programmeerimine põhikoolis, põhikooli informaatika õppekava

Summary

The heading: *RoboMind as an Attempt to Interest Middle School Teachers and Students in Programming.*

It can be assumed that the majority of Estonian Middle School students have no knowledge of educational programming even though the variety of programming environments designed for children has enhanced in time. In Estonia and in other countries the opportunity for children to learn programming is being more and more actively promoted foremost through learning Informatics. Educational programming is useful for children in many ways. Also, it gives a chance to feel the joy of making, designing and sharing something new and the basic knowledge and skills of informatics that can be useful later in higher education or in work.

The intent of present Bachelor's thesis was to create programming study material for Basic School students and to get feedback on the material and on the opportunity of educational programming from teachers and students. The study material consists of two worksheet sets and is meant approximately for two school lessons. The worksheet sets were tested by seven teachers and their students. The majority of teachers and students in this study found that educational programming may have a useful effect on children's future and on their ability to think. The majority of students assessed lessons interesting and easy and found RoboMind agreeable. However, the wish to learn or teach programming remained uncertain. According to some teacher's evaluation, the worksheets needed alteration so some changes in study material were made.

Keywords: programming in Basic School, Basic School informatics curriculum

Autorsuse kinnitus

Kinnitan, et olen koostanud ise käesoleva lõputöö ning toonud korrektselt välja teiste autorite ja toetajate panuse. Töö on koostatud lähtudes Tartu Ülikooli haridusteaduste instituudi lõputöö nõuetest ning on kooskõlas heade akadeemiliste tavadega.

Allkiri

Kuupäev

Kasutatud kirjandus

- Alice. Külastatud aadressil <http://dl.acm.org/citation.cfm?id=364161>
- Brauner, P., Leonhardt, T., Ziefle, M., & Schroeder, U. (2010). The Effect of Tangible Artifacts, Gender and Subjective Technical Competence on Teaching Programming to Seventh Graders. In J. Hromkovič R. Královič & J. Vahrenhold (Toim), *Teaching Fundamentals Concepts of Informatics* (pp. 61-71). Springer Berlin Heidelberg.
- Brusilovsky, P., Calabrese, E., Hvorecky, J., Kouchnirenko, A., Miller, P. (1997). Mini-languages: a way to learn programming principles. *Education and Information Technologies* 2(1), 65-83.
- Bundy, A. (2007). Computational thinking is pervasive. *Journal of Scientific and Practical Computing* 1(2), 67-69.
- Calder, N. (2010). Using Scratch: An Integrated Problem-solving Approach to Mathematical Thinking. *Australian Primary Mathematics Classroom*, 15(4), 9-14.
- Denner, J., Werner, L. (s.a.). *Measuring Computational Thinking in Middle School using Game Programming*. Külastatud aadressil <http://users.soe.ucsc.edu/~linda/pubs/AERAMeasuringCT.pdf>
- Denning, J. P. (2009). The Profession of IT beyond computational thinking. *Communications of the ACM* 52(6), 28-30.
- Diethelm, I., & Mittermeir, R. T. (Toim). (2013). *Informatics in Schools. Sustainable Informatics Education for Pupils of all Ages*. Springer Berlin Heidelberg
- Erahuvikool HuviTERA. Külastatud aadressil <http://ukuuraj.havike.eenet.ee/huvitera/ringid/#PROGEJAD> 27.12.2013
- Fesakis, G., Serafeim, K. (2009). Influence of the familiarization with "Scratch" on future teachers' opinions and attitudes about programming and ICT in education. *ACM SIGCSE Bulletin - ITiCSE '09* 41(3), 258-262.
- Feurzeig, W. (2010). Toward a Culture of Creativity: A Personal Perspective on Logo's Early Years and Ongoing Potential. *International Journal of Computers for Mathematical Learning* 15(3), 257-265. Springer Netherlands
- Gümnaasiumi riiklik õppekava (2011). Riigi Teataja I, 14.01.2011, 2.

- Gweon, G., Ngai, J. & Rangos, J. (2005). Exposing Middle School Girls to Programming via Creative Tools. In M. F. Costabile & F. Paternò (Toim), *Human-Computer Interaction - INTERACT 2005* (pp. 431-442). Springer Berlin Heidelberg
- Henderson, P. B., Cortina T. J., Wing, J. M., Hazzan, O. (2007). Computational Thinking. *ACM SIGCSE Bulletin* 39(1), 195-196.
- *HITSA Innovatsioonikeskus - ProgeTiigri programmi konstepstsioon*. Külastatud aadressil:
<http://www.innovatsioonikeskus.ee/sites/default/files/ProgeTiiger/ProgeTiigri%20kontseptsioon%2C%20veebilehele.pdf> 27.12.2013
- Hirsjärvi, S., Remes, P., & Sajavaara, P. 2004. *Uuri ja kirjuta*. Tallinn: Kirjastus Medicina.
- Hromkovič, J., & Steffen, B. (2011). Why Teaching Informatics in Schools Is as Important as Teaching Mathematics and Natural Sciences. In I. Kalaš & R. T. Mittermeir (Toim), *Informatics in Schools. Contributing to 21st Century Education* (pp. 21-30). Springer Berlin Heidelberg.
- *IKT hariduse populaarsus külastatud koolide õpilaste seas* (2012). Külastatud aadressil <http://itl.ee/?op=body&id=46>
- Ioannidou, A., Bennett, V., Repenning, A., Koh, K. H., Basawapatna, A. (2011). *Computational Thinking Patterns*. Külastatud aadressil
<http://files.eric.ed.gov/fulltext/ED520742.pdf>
- *JA Eesti õpilasfirma..* Külastatud aadressil <http://www.ja.ee/index.php?page=146&>
31.03.2014
- Kelleher, C. (2009). Supporting Storytelling in a Programming Environment for Middle School Children. In I. A. Iurgel, N. Zagalo & N., P. Petta (Toim), *Interactive Storytelling* (pp. 1–4). Springer Berlin Heidelberg
- *Kes õpetab 6.b klassile programmeerimist?* (2014). Külastatud aadressil
<http://uudiskiri.e-ope.ee/?p=13206> 04.05.2014
- *Kooliroboti projekt*. Külastatud aadressil
<http://www.robootika.ee/lego/projekt/index.php/projektist/> 27.12.2013
- Laanpere, M. (2010a). *Informaatika ainekava eesmärkidest ja ülesehitusest*. Põhikoolid valikaineraamat INFOTEHNOLOOGIA. Külastatud aadressil
http://www.oppekava.ee/index.php/Informaatika_ainekava_eesm%C3%A4rkidest_ja_%C3%BClesehitusest

- Laanpere, M. (2010b). *Kooliinformaatika eesmärkide ja sisu uuenemistrendid maailmas ja Eestis*. Külastatud aadressil http://www.oppekava.ee/images/b/bf/Kooliinformaatika_eesmärkide_ja_sisu_uuenemistrendid_maailmas_ja_Eestis.pdf
- Lee, Y. (2011). Scratch: Multimedia Programming Environment for Young Gifted Learners. *Gifted Child Today*, 34(2), 26-31.
- McInenrey, C. (2010). Having Fun with Computer Programming and Games: Teacher and Student Experiences. In J. Hromkovič, R. Královič & J. Vahrenhold (Toim), *Teaching Fundamentals Concepts of Informatics* (pp. 136-142). Springer Berlin Heidelberg
- Micheuz, P. (2011). A Competence-Oriented Approach to Basic Informatics Education in Austria. In R. T. Mittermeir & I. Kalaš (Toim), *Informatics in Schools. Contributing to 21st Century Education* (pp. 43-55). Springer Berlin Heidelberg
- *Programm ProgeTiiger*. Külastatud aadressil <http://www.progetiiger.ee/> 27.12.2013
- Põhikooli riiklik õppekava (2011). Riigi Teataja I, 14.01.2011, 1.
- Ratcliff, C. C., Anderson, S. E. (2011). Reviving the Turtle: Exploring the Use of Logo with Students with Mild Disabilities. *Computers in the Schools*, 28(3), 241-255.
- Repenning, A. (2012). Programming Goes Back To School. *Communications of the ACM* 55(5), 38-40.
- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., Kafai, Y. (2009). Scratch Programming for All. *Communications of the ACM* 52(11), 60-67.
- *RoboMind*. Külastatud aadressil <http://www.robomind.net/en/index.html> 30.03.2014
- Rohelaan, K. (2014, 17. märts). Noored ettevõtjad tahavad paremat IT-õpet. Äripäev, lk 8.
- Romeike, R. (2008). What's My Challenge? The Forgotten Part of Problem Solving in Computer Science Education. In R. T. Mittermeir & M. M. Syslo (Toim), *Informatics Education - Supporting Computational Thinking* (pp. 122-133). Springer Berlin Heidelberg.
- *Scratch*. Külastatud aadressil <http://scratch.mit.edu/projects/125297/> 26.03.2014
- Sengupta, P., Kinnebrew, S.J., Basu, S., Biswas, S., Clark, D. (2013). Integrating computational thinking with K-12 science education using agent-based computation:

A theoretical framework. *Education and Information Technologies*, 18(2), 351-380. Springer Berlin Heidelberg.

- Sentance, S., Dorling, M., & McNicol, A. (2013). Computer Science in Secondary Schools in the UK: Ways to Empower Teachers. In I. Diethelm & R. T. Mittermeir (Toim), *Informatics in Schools. Sustainable Informatics Education for Pupils of all Ages* (pp. 15-30). Springer Berlin Heidelberg
- Siegle, D. (2009). Developing Student Programming and Problem-Solving Skills with Visual Basic. *Gifted Child Today*, 32(4), 24-29.
- Syslo, M. M., & Kwiatkowska, A. B. (2013). Informatics for All High School Students. In I. Diethelm & R. T. Mittermeir (Toim), *Informatics in Schools. Sustainable Informatics Education for Pupils of all Ages* (pp. 43-56). Springer Berlin Heidelberg
- Tallinna Reaalkool. Külastatud aadressil
<http://www.reaalkool.ee/index.php/huviringid/robootikaring> 27.12.2013
- Tort, F., & Drot-Delange, B. (2013). Informatics in the French Secondary Curricula: Recent Moves and Perspectives. In I. Diethelm & R. T. Mittermeir (Toim), *Informatics in Schools. Sustainable Informatics Education for Pupils of all Ages* (pp. 31-42). Springer Berlin Heidelberg
- TTÜ Tehnoloogiakool. Külastatud aadressil
<http://www.ttu.ee/kooliopilasele/tehnoloogiakool/pohikooli-opilastele/> 26.03.2014
- Vilipõld, K., Antoi, K, Amitan, I. (2013). *Rakenduste loomise ja programmeerimise alused*. Valikkursus gümnaasiumitele. Õpik. Külastatud aadressil:
http://rlpa.ttu.ee/RLPA_opik.pdf
- Wing, J. M. (2008). Computational thinking and thinking about computing. *Phil. Trans. R. Soc. A* 336(1881), 3717-3725.
- Wolz, U., Stone, M., Pearson, K., Pulimood, S. M., Switzer, M. (2011). Computational Thinking and Expository Writing in the Middle School. *ACM Transactions on Computing Education (TOCE)* 1(2), Artikkel 9.

Lisad

Lisa 1. Esimene pöördumine õpetajate poole

Lugupeetud õpetaja!

Olen Tartu Ülikooli bakalaureuseastme üliõpilane ning õpin matemaatika- ja informaatikaõpetajaks. Kirjutan lõputööd, mille eesmärgiks on uurida matemaatika ja informaatika lõimimise võimalusi⁸ põhikoolis programmeerimise õppimise/õpetamise abil.

Pöördun Teie poole palvega, et leiaksite IV veerandi teisel kuni neljandal nädalal kahe koolitunni jagu, et 5.-9. klassi õpilastega (õpetaja enda otsustada on, millise klassiga oleks kõige parem sellist õppetööd läbi viia) arvutiklassis katsetada minu poolt koostatud õppematerjali. Olen teadlik, et uurimuse läbiviimiseks tuleb nõusolek saada koolidirektorilt, kuid sooviksin enne siiski teada, kas õpetajal oleks huvi ja aega, et kahe koolitunni jooksul veidi teist tüüpi mõtlemisülesannetega tegeleda. Bakalaureusetöö raames koostatud töölehtede eesmärgiks on tutvustada ja sissejuhatavalt õpetada õpilastele programmeerimist programmi RoboMind abil. Tegemist on haridusliku suunitlusega tasuta vabavaralise õppimiskeskkonnaga.

Samuti paluksin, et peale õppematerjali läbitöötamist täidaksid nii õpetajad kui ka õpilased eraldi tagasiside ankeedid õppematerjali huvitavuse, vajalikkuse, sobivuse ning asjakohasuse kohta. Töölehtedega töötamiseks tuleks õpetajal (vajadusel kooli IT-juhi abiga) esmalt arvutitesse paigaldada vastav programm (vajaliku rakenduse saab kätte sellelt leheküljelt - <http://www.robomind.net/en/download.html>) ning töölehed üle vaadata, et vajadusel õpilasi juhendada. Üldiselt on töölehed koostatud nii, et õpilased saaksid erinevas tempos samm-sammult iseseisvalt teadmisi omandada ja ülesandeid täita. Ankeedid ning õppematerjal on elektroonsel kujul kättesaadavad ning ankeete saab täita peale õppematerjaliga töötamise lõppemist vastava lingi kaudu. Lisaks pakuksin õpilastele võimalust ka kodus programmiga lisaülesandeid lahendada ja selle kohta vabatahtlikkuse põhimõttel tagasisideküsimustele

⁸ Siinkohal tuleb täpsustada, et algselt oli töö eesmärgiks uurida programmeerimist kui informaatika ja matemaatika õpetamise lõimimise võimalust, kuid hiljem töö eesmärk muutus ning programmeerimise õpetamise eesmärgid ei pea koolis ilmtingimata olema seotud matemaatikaõpetuse eesmärkidega.

vastata.⁹ Uurimuses osalevate õpetajate ja õpilaste nimed ei kajastu lõputöös. Palun õpetajal, kes on uurimuses osalemisest huvitatud, kirjutada mulle kaitootsi@hotmail.ee, et saaksime alustada koostööga. Võib aimata kui raske on ainekava välistele asjadele kahe koolitunni jagu aega kulutada, kuid kui mõni õpetaja siiski leiaks selle aja, siis olen abi eest väga tänulik.

Lugupidamisega

Kai Tootsi

⁹ Siinkohal tuleb märkida, et lisäülesannete kodus lahendamist siiski üheski klassis ei toimunud.

Lisa 2. Teine pöördumine õpetajate poole

Tere!

Olen väga tänulik, et võitsite vaevaks minu koostatud töölehti katsetada. Vabandan, et töölehtede saatmisega aega läks. Saadan ülevaate, kuidas võiks meie koostöö edaspidi sujuda.

TÖÖLEHED

Kirja manuses on pdf-failina kahe tunni materjal programmeerimise õppimiseks/õpetamiseks programmi RoboMind abiga. Seega eelnevalt peab arvutites olemas olema programm RoboMind (leiate selle siit <http://www.robomind.net/en/download.html>).

Oleksin väga tänulik, kui jõuaksite selle või järgmise nädala jooksul töölehti katsetada. Materjalid on koostatud programmi manuaali ja teiste programmide juhendite toel. Kuna ei olnud otsest eeskuju teiste töölehtede näol, kuidas selle teema kohta õppematerjali edukalt koostada, siis olen teinud töölehed ülikoolis õpetatuga sarnase struktuuriga. Materjali raskus võib seega olla klassiti erinev.

MIDA PEAKS ÕPETAJA JÄLGIMA

1. Tundide materjal ei ole täpselt ajaliselt mõõdetud. Seega võib ajast puudu tulla ning võimalik, et mõnes klassis ka aega üle jääda. Sellega saab õpetaja veidi arvestada, vaadates üle materjalid ja teades enam-vähem oma klassi taset.
2. Kui aega tuleb puudu, siis võibki tund selle koha peal pooleli jääda ning jätkata saab järgmisel tunnil. Kui ka teise tunniga ei jõuta materjali läbi töötada, siis saab õpetaja tagasisides selle ära märkida. Kui aega peaks üle jääma, võib vajadusel ka teise tunni materjali õppimisega alustada.
3. Õpetaja saab abiks olla sellega, et õpilased programmi leiaks ja selle avada saaks.
4. Materjalis õpetatakse oma töid salvestama ja hiljem ka avama. Siin saab õpetaja läbi mõelda, kuhu täpsemalt töid võiks salvestada.
5. Materjalis on esitatud õpilastele küsimusi. Õpetaja valida on, kas küsimused arutatakse koos läbi, arutlevad õpilased lauaaabriga või mõtleb igaüks ise küsimused enda jaoks läbi.
6. Õpetaja saab kontrollida klassis, kas õpilased saavad hakkama materjaliga (pdf-kujul) töötamisega.
7. Samuti võiks õpetaja võimaluse korral jälgida, kas kõik õpilased saavad tagasiside ankeedid täidetud ja saadetud.

TUNNIKÄIK

Tunni materjal on koostatud nii, et õpilane peaks ise tunni materjali lugema ja paralleelselt töötama programmiga.

Kui tundide materjal saab läbi töötatud, siis on õppematerjali lõpus toodud õpilastele vastav viide ankeedile, mille peaks jõudma ära täita teise tunni lõpus.

Olen arvestanud, et täitmiseks võiks kuluda ~15 minutit.

KOOSTÖÖ

Igasuguste probleemide, tõrgete ning ehk ka edu korral võib mulle kirjutada siia meilile. Loen seda aktiivselt iga päev ja saan ehk murede korral abiks olla.

TAGASISIDE

Kui õpetajad on tundide läbitöötamise lõpetanud, siis palun kirjutada mulle, et saaksin saata tagasiside andmiseks viite vajalikule ankeedile. Olge kenad ja kirjutage mulle, millal te planeerite tunnid läbi viia, et teaksin arvestada.

Lugupidamisega ja heale koostööle lootes

Kai Tootsi

Lisa 3. Elektrooniline ankeet õpetajatele

Programmeerimise õpetamine põhikoolis

Küsitluse täitmise ajaks olete läbi viinud kaks tundi programmeerimise õpetamiseks programmi RoboMind abil. Nüüd palun, et avaldaksite arvamust antud töölehtede, programmeerimise õppimise ja õpetamise kohta. Selleks palun vastata järgnevatele küsimustele. Vastamise käigus jääte Te anonüümseks. Vastamisele võib kuluda kuni 30 minutit.

*1. Teie sugu?

☐ Naine

☐ Mees

☐ 5. klass

☐ 6. klass

☐ 7. klass

☐ 8. klass

☐ 9. klass

☐ Matemaatika

☐ Informaatika

☐ Füüsika

☐ Muu

Siin võib valida ka mitu vastusevarianti.

*4. Kua olete õpetajana töötanud?

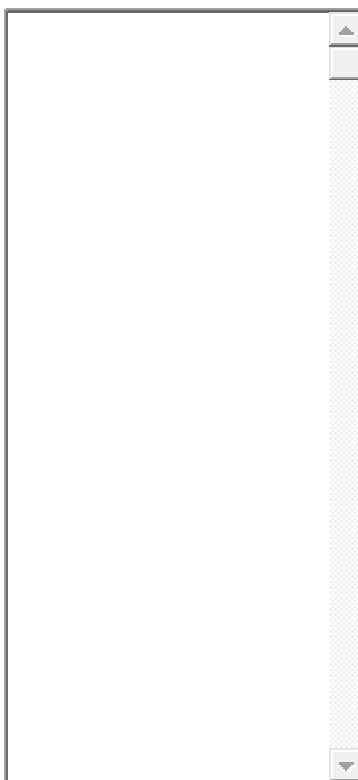
Palun vastata arvuliselt.

*5. Kas olete ka varem programmeerimist õpetanud koolis?

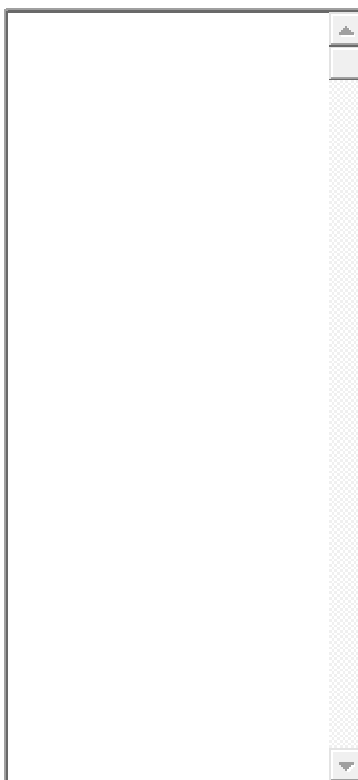
☐ Jah ☐ Ei

Kui vastasite "Ei," palun liikuda edasi 9. küsimuse juurde.

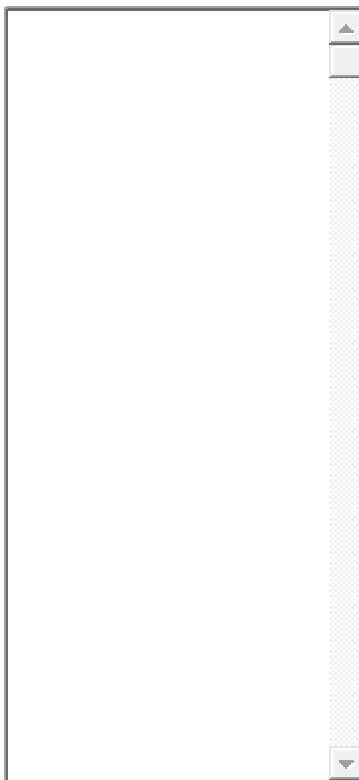
6. Palun kirjeldage, kuidas ja mil määral olete Te programmeerimist õpetanud?

A large, empty rectangular text box with a thin black border. On the right side, there is a vertical scrollbar with a small upward-pointing triangle at the top and a downward-pointing triangle at the bottom.

7. Milliste programmide abil olete Te programmeerimist õpetanud?

A large, empty rectangular text box with a thin black border. On the right side, there is a vertical scrollbar with a small upward-pointing triangle at the top and a downward-pointing triangle at the bottom.

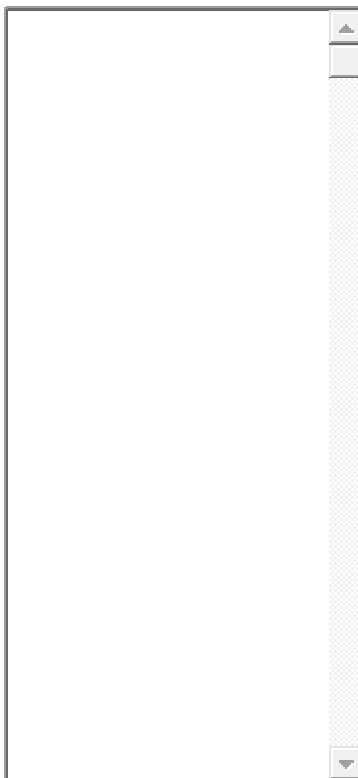
8. Kas ja mil määral kavatsete Te ka edaspidi koolis seda teemat käsitleda?



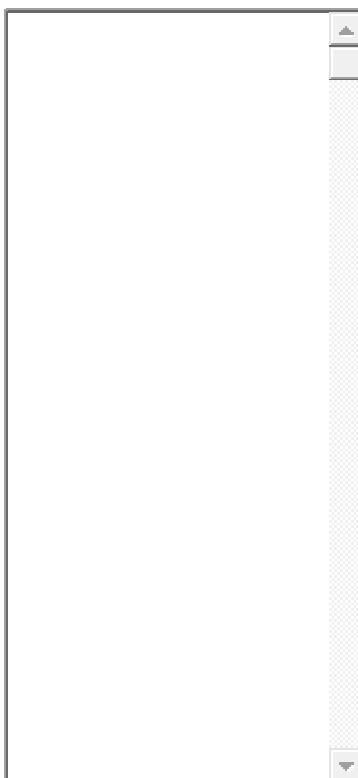
9. Kui vastasite 4. küsimusele "Ei," siis kas kaalute tulevikus programmeerimise õpetamist?

- ☐ Jah, kindlasti
- ☐ Võimalik jah
- ☐ Ei oska öelda
- ☐ Pigem mitte
- ☐ Kindlasti mitte

10. Palun põhjendage eelmisele küsimusele antud vastust.

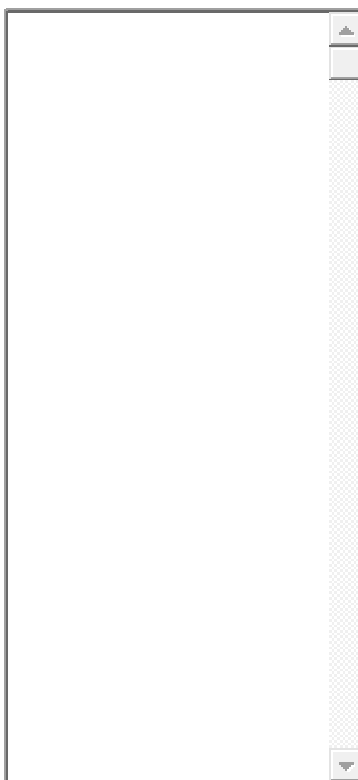
A large, empty rectangular text input field with a thin grey border. On the right side, there is a vertical scrollbar with a small upward-pointing arrow at the top and a downward-pointing arrow at the bottom.

*11. Kust olete teadmisi saanud programmeerimise kohta?
Kui ei ole eelnevaid teadmisi selle teema kohta, siis palun
vastake nii.

A large, empty rectangular text input field with a thin grey border. On the right side, there is a vertical scrollbar with a small upward-pointing arrow at the top and a downward-pointing arrow at the bottom.

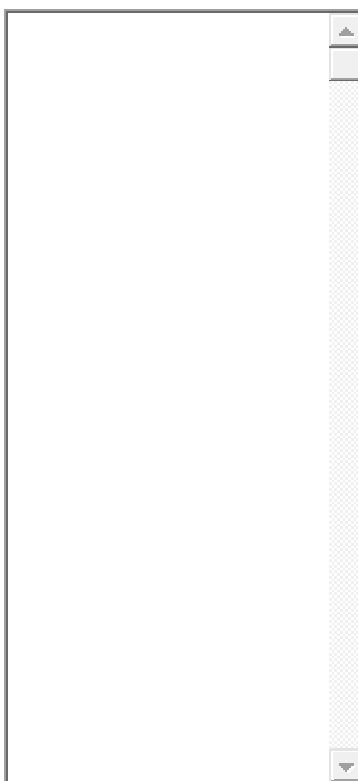
*12. Miks peaks põhikooli õpilastele õpetama programmeerimist?

Kui arvate, et ei peakski, vastake nii ja palun põhjendage.



Kui arvasite, et seda teemat ei peaks käsitlema, siis liikuge palun 16. küsimuse juurde.

*13. Millistele klassidele ja mil määral sobiks programmeerimist õpetada?



*14. Milliste ainete õpetajad võiksid õpetada lastele programmeerimist?

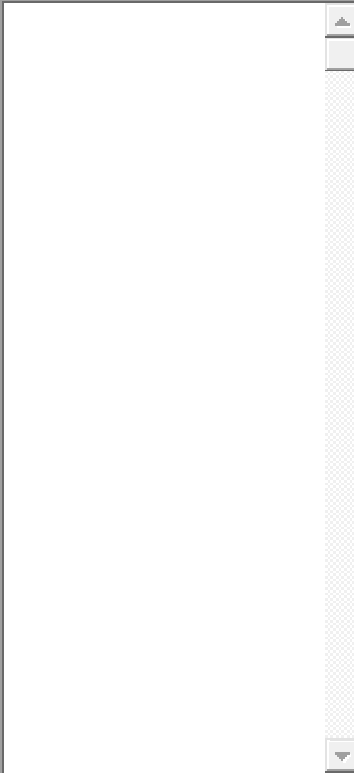
- ☐ Matemaatika õpetaja
- ☐ Füüsika õpetaja

☐ Informaatika õpetaja

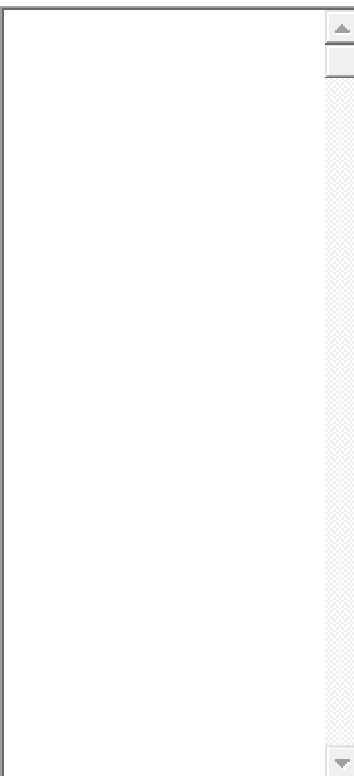
☐ Muu aine õpetaja

Siin võib valida ka mitu vastusevarianti.

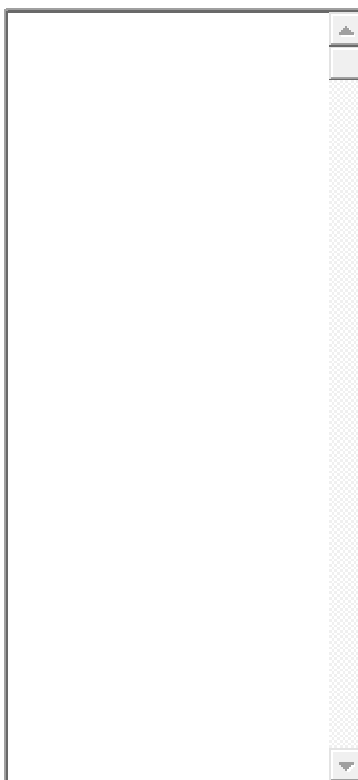
*15. Kas ja kuidas näete programmeerimist põhikoolis õpetatava ainena või teemana?

A vertical rectangular text input field with a thin grey border. On the right side, there are two small square buttons: the top one has an upward-pointing arrow, and the bottom one has a downward-pointing arrow.

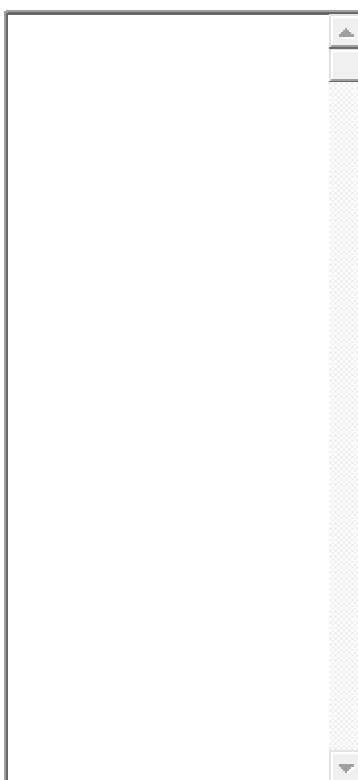
*16. Kas ja kuidas näete programmeerimist matemaatika ja informaatika lõimimise võimalusena?

A vertical rectangular text input field with a thin grey border. On the right side, there are two small square buttons: the top one has an upward-pointing arrow, and the bottom one has a downward-pointing arrow.

*17. Milliseid oskusi või teadmisi võiks programmeerimine õpilastel arendada?

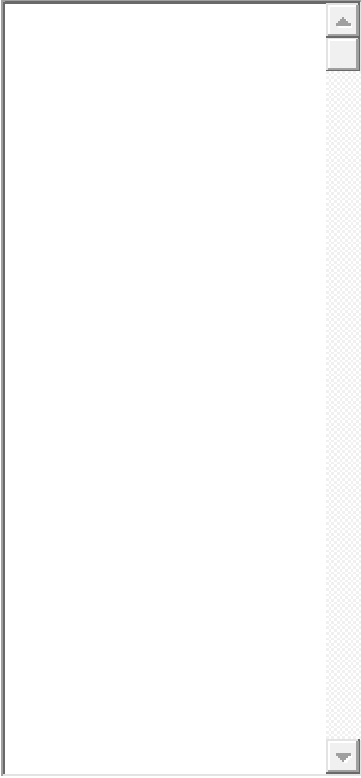
A large, empty rectangular text box with a thin black border. On the right side, there is a vertical scrollbar with a small upward-pointing arrow at the top and a downward-pointing arrow at the bottom.

*18. Kuidas sobis valitud programm RoboMind programmeerimise õpetamiseks Teie klassile?

A large, empty rectangular text box with a thin black border. On the right side, there is a vertical scrollbar with a small upward-pointing arrow at the top and a downward-pointing arrow at the bottom.

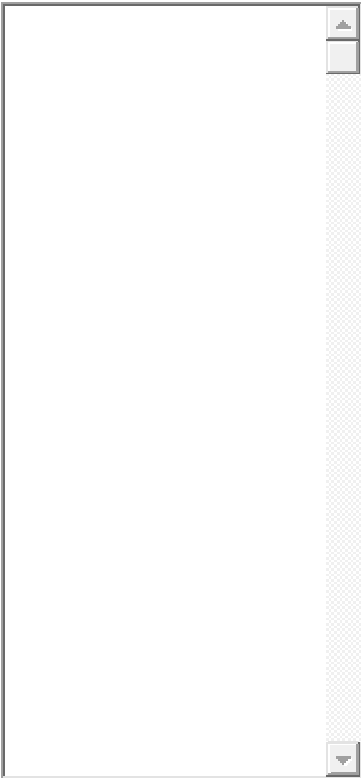
Palun hinnake vabas vormis programmi sobivust (häid/halbu omadusi).

*19. Kuidas sobisid koostatud töölehed antud teema tutvustamiseks?

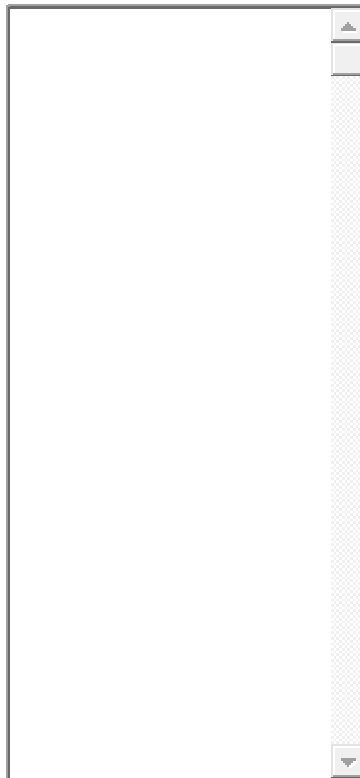
A large, empty rectangular text box with a thin black border. On the right side, there is a vertical scrollbar with a small upward-pointing arrow at the top and a downward-pointing arrow at the bottom.

Palun hinnake vabas vormis materjali sobivust õppimiseks/õpetamiseks.

*20. Palun tooge välja, milliseid programme üldse olete matemaatika- või informaatikaõpetuses kasutanud?

A large, empty rectangular text box with a thin black border. On the right side, there is a vertical scrollbar with a small upward-pointing arrow at the top and a downward-pointing arrow at the bottom.

*21. Millisel viisil viisite need tunnid läbi?



Palun tooge välja, mil määral te rääkisite infot juurde või täiendasite koostatud slaide.

Lisa 4. Elektrooniline ankeet õpilastele

Sinu arvamus tunnist ja programmeerimisest

Võtsid osa just programmeerimise õppimisest. Nüüd soovin teada, kuidas Sul see tund läks ja mida arvad programmeerimise õppimisest. Püüa vastata küsimustele nii ausalt kui võimalik. Ära pelga kirjutada, kui tund oli liiga kerge või ka liiga raske. Sinu arvamus on oluline! Ankeet on anonüümne, see tähendab, et Sinu nimi ei ole kellelegi hiljem teada.

*1. Sugu:

- ☒ Mees
- ☐ Naine

*2. Mitmendas klassis Sa õpid?

- ☐ 5. klassis
- ☐ 6. klassis
- ☐ 7. klassis
- ☐ 8. klassis
- ☐ 9. klassis

*3. Kas oled koolis õppinud informaatikat või arvutiõpetust?

- ☐ Jah
- ☐ Ei

*4. Kas olid enne seda tundi kuulnud midagi programmeerimisest või sellega kokku puutunud?

- ☐ Jah
- ☐ Ei

5. Kui vastasid eelmisele küsimusele jaatavalt, siis kust või kuidas oled kuulnud midagi programmeerimisest?

*6. Mis Sa arvad, millised asjad tavaelust võivad olla tekkinud just tänu programmeerimisele?

Siia võib vabas vormis panna kirja erinevaid asju.

*7. Kas tahaksid õppida koolis programmeerimist?

- ☐ Kindlasti mitte
- ☐ Pigem mitte
- ☐ Võib-olla
- ☐ Pigem jah
- ☐ Jah tahaksin

*8. Palun põhjenda eelmise küsimuse vastust:

*9. Kuidas hindad nende kahe tunni huvitavust?

Siin võib valida ka mitu vastusevarianti.

- ☐ Huvitav oli
- ☐ Ei oska hinnata
- ☐ Ebahuvitav oli

*10. Kuidas hindad nende kahe tunni sisu mõistmist?

- ☐ Liiga raske oli
- ☐ Piisavalt raske oli
- ☐ Igav oli

☐ Lihtne oli

*11. Miks peaks koolis programmeerimist õppima?

Siin võib valida ka mitu vastusevarianti.

☐ Äratab huvi reaalainete vastu

☐ Äratab huvi tehnika vastu

☐ Äratab huvi arvuti vastu

☐ Äratab huvi matemaatika vastu

☐ Äratab huvi füüsika vastu

☐ Läheb edaspidi elus vaja

☐ Annab kasulikud teadmised keskkoolis või
ülikoolis õppimiseks

☐ Õpetab loogiliselt mõtlema

☐ Annab arusaamise masinate tööst

☐ Ei peakski õppima

☐ Muu põhjus

12. Kui valisid eelmise küsimuse vastuseks "Muu
põhjus," siis palun too see siin välja:



*13. Kas oled kuulnud midagi robotikast?

☐ Jah

☐ Ei oska öelda

☐ Ei

*14. Kas tahaksid õppida koolis robotikat?

☐ Kindlasti mitte

☐ Pigem mitte

☐ Võib-olla

- ☐ Pigem jah
- ☐ Jah tahaksin

*15. Kuidas Sulle meeldis programm RoboMind?

- ☐ Üldse ei meeldinud
- ☐ Pigem ei meeldinud
- ☐ Ei oska öelda
- ☐ Pigem meeldis
- ☐ Väga meeldis

*16. Palun põhjenda eelmise küsimuse vastust:

Suur aitäh Sulle!

Lisa 5. Töölehtede komplekt I



1. TUND

Programmi looja: Arvid Haima, Amsterdami Dlikool

Selle tunni jooksul:

- Õpime tundma programmi RoboMind ja teeme esimesi samme programmeerimise suunas
- Selleks uurime programmi slaididel olevate juhiste kaudu, täidame ülesandeid ja vastame küsimustele
- Programm on inglisekeelne
- Küsida õpetaja abi, kui ei saa mõnest sõnast aru!

Loodame, et see tund saab olema uudne ja huvitav kõigile!

Mis on RoboMind?

- Programmeerimise keskkond, mis on mõeldud harjutamiseks alates 2. klassist kuni gümnaasiumini
- Pakub teadmisi loogikas, informaatikas ja robotikas
- Selle vahendi abil saab õppida ka programmeerimist

Mis võiks olla aga programmeerimine?

Milleks on mõeldud käsud?

- Selleks, et panna masinat (ka arvutit või robotit) midagi tegema, peame esmalt andma juhised
- Igal masinal on olemas käskude kogum, mida see suudab täita
- Näiteks mõnel pesumasinal on erineva pesu jaoks erinev programm, mis nõuab erinevat vee temperatuuri, pesemisaega või tsentrifuugimise taset
- Peaaegu igale nupule mingi masina küljes vastab käskudekogum, mida see peab täitma

Arvuti programmeerimine

- Näiteks on võimalik koostada programm, mille kohaselt arvuti peab end mõne aja pärast välja lülitama. Inimene ei pruugi samal ajal arvuti taga isegi olla.

Mõtlen oma koduarvuti peale - võib juhtuda, et kui mõnda aega arvuti taga mitte olla, siis läheb ekraan mustaks või läheb arvuti unerežiimile

Me ei vajuta kuhugi, aga ometi see juhtub? Miks?

Mida teeb programm?

- Programm juhib arvutit nii nagu inimene on selleks juhised andnud
- Programmeerimine on programmide kirjutamine käskude keeles, millest arvuti aru saab
- Programmeerides peab kõik olema hästi läbimõeldud, et arvuti täidaks meie käsked täpselt nii nagu soovinud oleme



RoboMind

- RoboMindi tegelaseks on **robot**
- Teda saab nii puldi abil juhtida kui ka programmeerida
- Me hakkame talle ütlema, mida ta peab tegema
- Ehk me hakkame talle **käsk** jagama – teda **juhtima**
- Tema juhtimiseks tuleb kirjutada valmis **kood**, mis sisaldab käsk

7



Aga alustame siis..

- Avame programmi RoboMind
- Uurime avanenud akent. Mida me siin näeme?
- 1. Silma peaks hakkama **robot** (ühe ruudu sees)
- 2. Üleval nurgas on **kaart** (rist tähistab robotit)
- 3. Tumedasse aknasse tulevad **käsk**, mida robot hakkab täitma
- 4. Kui me oleme käsk valmis kirjutanud, käivitame koodi (▶ nupp –käivitamise nupp - leidsid üles?)

Vaata, kas näed ekraanil kõiki neid asju? Proovi ka ▶ nuppu vajutada. Mis juhtus?
Kuhu aknasse ilmub arvuti vastus?

9



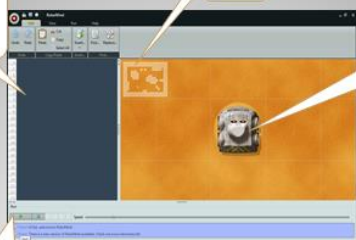
Mis on kus?

Käsk

Kaart

Robot

Käivitamine



9



Veateated

- Arvuti pidi andma ▶ nupuvajutuse peale veateate - **Error on line 1: There is no script to run**
- **Line 1** – tähendab, et viga on esimesel real
- **No script to run** – tähendab, et ei ole kirjutatud käsk, mida käivitada
- Arvuti (või ka robot ise) peabki andma veateate, kui mingil põhjusel ei sobi sisestatud käsk täitmiseks
- Vead, mis võivad tekkida programmeerimisel:
 - Kirjavead* – näiteks jääb mõni täht kuskilt ära
 - Käsuvead* – näiteks arvuti sellist käsku ei tunne
 - Muud vead* – näiteks ei mahu robot nii palju liikuma, kui on soovitud (siis annab samasse aknasse teate robot ise)

11



Pilt suuremaks või väiksemaks

- Et saada paremat ülevaadet, võib RoboMindi akent muuta suuremaks või väiksemaks
- Selleks leiame ülevalt menüü **View** ja sealt nupud **Zoom in** või **Zoom out**
- Kui vajutada neid nuppe, siis muutub vaade robotile ülevaatlikumaks või kitsamaks

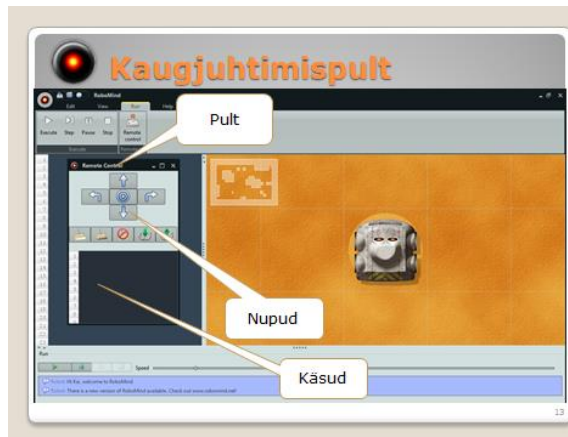
11



Mida saame robotiga teha?

- Robotit saab juhtida kahel viisil: **puldi** või **käskude kirjutamisega**
- Puldi abil juhime me robotit koheselt
- Teine võimalus on enne käsk valmis kirjutada ja siis need **käivitada**. Nii saab robotit panna tegema mitmeid asju korraga ilma et peaks selleks pidevalt pulti vajutama
- Kuid proovime ka pulti!
- **Avame puldi** – selleks avame ülevalt menüü **Run** (**käivita**) ja sealt **Remote Control** (**pult**)
- **Vaata järgmist slaidi. Kas Sul on samasugune aken ees?**

12



Puldiga töötamine

- Nüüd avanes puldi aken, kus on **nupud** ja all tumedalt ka **käsurida** (koht, kuhu ilmuvad käsud)
- Liigu hiirega aeglaselt erinevate nuppude peal, kuid ära vajuta neid veel!
- Piisavalt aeglaselt liikudes on näha, et iga nupu jaoks on olemas ka käsk (N: nupule „Üles” vastab käsk **forward(1)**, mis tähendab *edasi 1 koha võrra*)

Nüüd on aeg katsetada nuppe! Teeme seda!

Proovi erinevaid nuppe. Mida saab robotiga teha?

- Kas proovisid värvida? Erinevaid värve?
- Kas üritasid tõsta midagi ja maha panna? Kas robot andis ka selle peale mingi teate vastavasse aknasse?
- Kas leidsid nupu, mis viib roboti algasendisse tagasi?
- Kui ei, siis proovi kõiki neid asju teha.

Käskude täitmine

- Käske peaks robot täitma kohe
- Kuid võib tekkida ka olukord, kus robot seda käsku täita ei saa
- Siis ilmub programmi helesinisesse aknasse mingi teade
- Näiteks, kui lasta robotil väga palju ühes suunas liikuda, siis teatud hetkel jõuab ta seinani.
- Kui ta seinani on jõudnud ja ikka anda käsk liikumiseks, siis annab robot veateate aknasse

Proovi, kas Sul ka nii juhtub?

Kui oleme robotiga mängimise lõpetanud..

- Nuppe vajutades nägime, et musta aknasse ilmusid kordamööda käsud, mida robot täitis
- Igal nupule vastab oma käsk:
 - `backward(1)` – robot liigub ühe koha võrra tagasi
 - `forward(2)` – robot liigub kahe koha võrra edasi
 - `left()` – robot pöörab end vasakule
 - `right()` – robot pöörab end paremale
 - `paintWhite()` – robot paneb maha valge pintsli
 - `paintBlack()` – robot paneb maha musta pintsli
 - `stopPainting()` – robot võtab pintsli üles
 - `pickUp()` – robot võtab maast asja
 - `putDown()` – robot paneb võetud asja maha

Käskude kirjutamise eelised puldi ees

- Selleks, et robot teeks mingi liigutuse, pidime me vajutama eelnevalt nuppu
- Iga liigutuse jaoks pidi vajutama nuppu. Pikapeale võib minna tüütuks?
- Kuid robotit on võimalik korraga panna mitut käsku järjest täitma
- Ja seda ilma pidevalt nuppudele vajutamata!

Mis Sa arvad, kuidas? Katseta!



Käsud?

1. Siia kirjuta käsud

2. Siit käivita käsud



3. Siin vaata, mis juhtub


19



Teeme esimese katse..

• Et käske sisestada:

- 1) Võime need ise kirjutada käsuraale
- 2) Või valida soovitud käsud menüüst **Insert...**

Katseta mõne käsu valimist menüüst **Insert...** ja proovi kohe ka roboti peal (kasuta selleks käivitamise nuppu ).

Kas robot toimis nii nagu pidi?

Proovi robotit mitut asja tegema panna.

20



Esimene ülesanne

• Kirjutame järgmised käsud käsuraale:

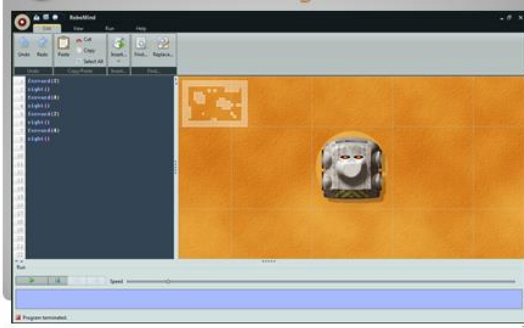
```
forward(2)
right()
forward(4)
right()
forward(2)
right()
forward(4)
right()
```

Piilu ka järgmist slaidi ja kontrolli kas Sul õnnestus ka nii?
Kas robot jõudis algasendisse tagasi? Miks? Millist kujundit mõõda robot sõitis?

21



See näeks välja nii..



22



Kuidas oma tööd salvestada?

• Kui tahta hiljem oma käske üle vaadata:

1. Selleks leida üleval programmi vasakul nurgas üles „roboti silm“
2. Vajutada sellele – avaneb menüü
3. Leida üles käsk „**Save**“
4. Leia endale kaust, kuhu soovid oma käsujada salvestada
5. Pane oma tööle **nimi**
6. Vajuta nuppu „Salvesta“

Jäta meelde, kuhu oma tööd salvestasid ja mis nimega, et neid hiljem avada!

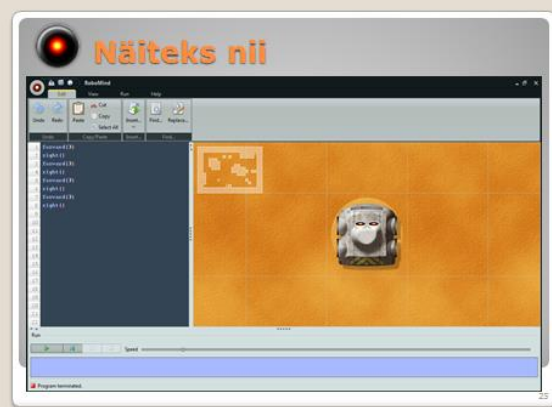
23



Teine ülesanne

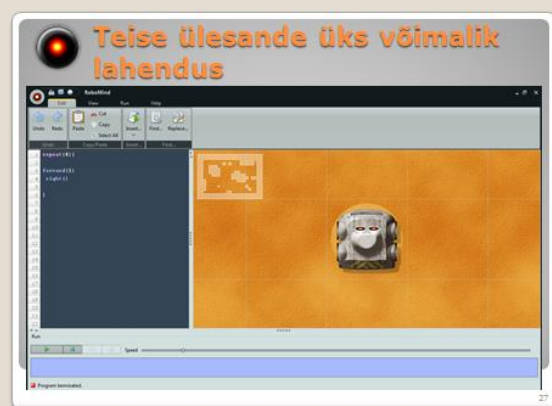
- Proovime roboti nii liikuma panna, et ta sõidaks **ruudu** kujulist joont mööda ja tuleks oma kohale tagasi
- Selleks kirjuta käsuraale vajalikud käsud ja käivita need
- Käskude loomisel tuleb arvestada ka **takistustega!**
- Vaata, mis juhtub, kui roboti teele peaks jääma takistus
- Näita õpetajale oma lahendust!

24



Seda ülesannet saab lahendada ka nii

- Ka vähema arvu käskudega saab sama asja saavutada
- Selleks on olemas kasulik käsk: `repeat(arv){juhised}`
- Arvuga ütleme, mitu korda kordame antud juhiseid
- Juhiste koha peale kirjutame soovitud käsud, mida robot peab tegema
- Otsi see käsk **Insert...** menüüst üles!
- Proovi ise selle käsu abil teist ülesannet teha!
- Kui jääd hätta, piilu järgmist slaidi.



Kolmas ülesanne: `repeat(arv){juhised}`

- Ruutu me juba proovisime joonistada, proovime riskülikut ka.
- See ülesanne võib olla mõnevõrra keerulisem..
- Põhiline on ise programmi kirjutamisel see, et ei kardaks proovida.
- Kui mõni käsk on vale või ei ole päris kindel, kuidas asi toimib, saadki teada **katsetades**.
- Kõigepealt katseta, kas saad ise tehtud. Slaididel olevad käsud ei pruugi kattuda Sinu omadega, sest igal ülesandel võib olla **mitu lahendust!**
- Kui ikka üldse ei õnnestunud, piilu järgnevat slaidi ja proovi siis teha..



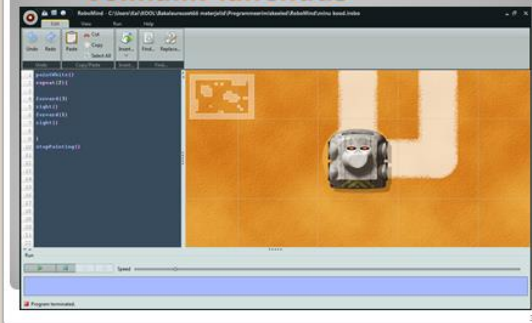
Hakkame värvima!

- Robotile saab enne liikuma hakkamist anda ka käsu *pintsel maha panna* - `paintWhite()` VÕI `paintBlack()`
- Otsi need käsud **Insert...** menüüst üles
- Proovi muuta käskude rida nii, et joonistaksid ruudu küljed mõlema värviga

Proovi ka riskülikut joonistada!

Näita õpetajale oma lahendusi!

Värvimise ülesande üks võimalik lahendus



31

Proovime nüüd ka koostatud ülesannete lahendusi avada

1. Selleks leia üleval programmi akna vasakul nurgas üles „roboti silm“
2. Vajuta sellel – avaneb menüü
3. Leia üles käsk „Open“
4. Leia üles kaust, kuhu oma tööd salvestasid
5. Ava mõni töö ja proovi seda uuesti käivitada

Kas kõik õnnestus? Kui ei, kutsu õpetaja appi või proovi uuesti.

32

Materjali pani kokku
Kai Tootsi

Altäh!

35

Lisa 6. Töölehtede komplekt II



2. TUND

Programmi looja: Arvid Haima, Amsterdami Dilekool

1



Selle tunni jooksul

- Õpime tundma veelgi käske, mida robot suudab täita
- Vaatame, mida tähendavad tingimusi sisaldavad käsud
- Võtame kokku, mida õppisime
- Vastame küsimustikule

2



Väike meeldetuletus

Eelmises tunnis:

- Õppisime robotiga liikuma, värvima, asju üles korjama
- Õppisime pulti tundma
- Õppisime käsureale ise käske kirjutama
- Õppisime ära kasuliku käsu `repeat(arv){juhised}`
- Õppisime töid salvestama ja avama

Kas Sul on see kõik meeles?

Kui ei ole, piilu eelmise tunni slaide!

3



Avame programmi

Ja avame menüü **Insert...** – millised valikud meil on?




4



Move ehk liikuma



5



Liikumise käsud

- Me teame juba käske:
`forward(1)`, `backward(1)`, `left()`, `right()` – kusjuures **1** asmele võib panna ka mõne teise arvu!
N: `forward(3)` korral liigub robot lihtsalt **3** koha võrra edasi
- Uued käsud:
`north(1)`, `south(1)`, `east(1)`, `west(1)` – ka siin võib **1** asemele panna mõne teise arvu
N: `west(6)` korral liigub robot **6** kohta lääne suunas

Proovi liigutada robotit erinevatesse ilmakaartesse erinevate arvude korral!

6



Tingimuslaused

- Uurime ka **Conditions>** menüüd
- Siin on käsk `if(tingimus){tegevus}`
- **if** tähendab *kui*
- kui **tingimus** on täidetud, siis teeb robot **tegevuse**
- See käsk aitab panna robotit **tegema midagi** nii kaua, kui **tingimus** (mis me ise määrame!) on täidetud

7



Tingimuslaused



8



Näited

- Näiteks igapäevaelus inimene ei saa enne autoga sõitma hakata, kui ta ei ole süüdet sisse keeranud
- Kui selle saaks panna tingimuslausena kirja, siis võiks nii:
`if(süüde sisse keeratud){hakka sõitma}`
- Seega tingimus peab olema täidetud - enne ei saa tegevust täita
- VÕI enne ei lõppe tegevus - näiteks `if(ülekäigurada ees){auto seisab}`

Robotile võib ette anda näiteks järgmise tingimuse käsu:

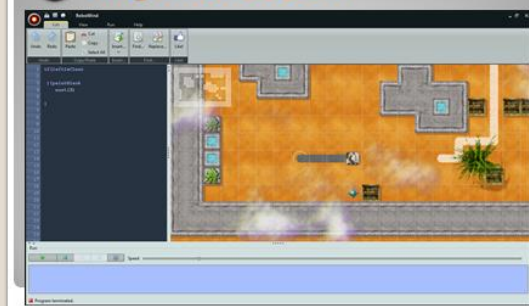
`if(frontIsClear){forward(1)}` - kui ees on ruut vaba, siis liigu 1 kohta võrra edasi!

Sisesta see käsk ja vaata, mis juhtub! Proovi muuta käsku!

9



Tingimuslause if



10



Veel tingimustest..

Veidi teistmoodi käsk:

`if(tingimus){tegevus1}else{tegevus2}`

- **else** tähendab *teine* või *muu*
- kui **tingimus** on täidetud, siis teeb robot **tegevuse 1** kui ei ole täidetud, siis teeb **tegevuse 2**

Igapäevaelu tingimuslausena:

`if(süüde sisse keeratud){hakka sõitma}else{keera süüde sisse}`

Katseta ka teisi **Conditions>** menüü käske!

11



Ülesanded

1. Kirjuta käsud nii, et kui maapind roboti ees on tühi, siis tuleb robotil see valgeks värvida ja liikuda ühe võrra edasi. Käivita ja vaata!
2. Kirjuta käsud nii, et kui maapind robotist vasakul on tühi, siis tuleb robotil keerata vasakule ja värvida 3 ruttu mustaks.
3. Kirjuta käsud nii, et robot liiguks *märgutule*(beacon) juurde ja sööks selle ära.

12



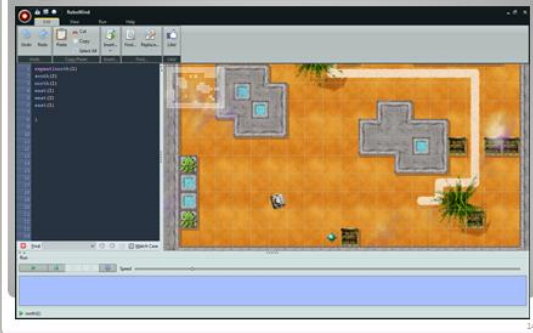
Loops ehk tsüklid

- Tsükkel on sündmuste ring ehk *kordamine(repeat)*.
- Tsükkel võib teinekord anda sama tulemuse mis väga pikk käskude rida
- Näiteks katseta käsku `repeat{forward(1)}`
- Proovi kirjutada ka käsk `repeat{forward(1) backward(1)}`
- Selleks, et tegevus lõpetada, tuleb vajutada punast **stop** nuppu.
- Ülesanne: kasuta tsüklit, et robot igas suunas tantsima panna!

13



Näide



14



Tsüklid

- Selleks et tsükli rakendust leida ilma et tekiks lõputu kordamine, saab seda käsku täiendada teiste käskudega
- Loops> menüüst leiab ka käsu `repeat(3){tegevus}`
- 3 näitab, mitu korda `tegevust` tehakse
- Numbrit saab muuta!
- Ülesanne: Kirjuta käsud kasutades tsüklit nii, et robot joonistaks ruudu

Vihje: `{tegevus}` võib sisaldada mitut käsku, kui need üksteisest eraldada

15



Tsüklid

- Käsk `repeatWhile(tingimus){tegevus}`
- Ehk Korda `tegevust` kuni `tingimus` kehtib
- Näiteks käsul `repeatWhile(leftIsClear){left}` jääbki `tingimus` kehtima. Uuri järgi!
- Kas ka siinjuhul - `repeatWhile(leftIsClear){left forward(1)}` jääb tegevus lõputult korduma?
- Ülesanne: Kirjuta käsud kasutades tsüklit nii, et robot liiguks diagonaalis (1 ruut edasi ja 1 ruut kõrvale) niikaua kuni tuleb sein ette

16



Tingimuslaused ja tsüklid

- Tsükli sees võib kasutada ka tingimuslause käsku
- Näiteks
`repeat(){if(frontIsObstacle()){left()}}else{forward(1)}`
- kui ees on takistus, tuleb pöörata vasakule, kui ei ole takistust, liikuda edasi 1 võrra
- Ja korrata tegevust koguaeg
- Katseta!
- Siinkohal on kasulik valida menüüst **View** nupp **track robot**.

17



Käsud tsüklist väljaspool

- Tsüklist väljaspool võib samuti kasutada käske. Tsükkel võib olla ainult üks osa kirjutatud käskudest.
- Näiteks
`paintWhite`
`repeat(3){south(1) north(2) east(1) west(2)}`
- Näiteks
`repeatWhile(frontIsClear){south(1) east(1)}`
`frontIsBeacon`
`eatUp`

18



Kokkuvõtteks

- RoboMind'i abiga võisime lihtsal kujul õppida programmide kirjutamist
- Ise kirjutasime sellised programmid, mis aitaks meil robotit enda soovi kohaselt liigutada
- Selliseid RoboMind'ga sarnaseid abivahendeid on veelgi...
- Programmeeritakse roboteid, kasse, kilpkonni ja muud
- Liigutatakse, joonistatakse, värvitakse, tehakse häält...
- Palju on veel, mida võiks avastada ja õppida!

19



Kokkuvõtteks

- Robotiga mängimine on alles programmeerimise algus
- See, millega firmades tegeletakse, on palju keerulisem ja nõuab rohkem teadmisi ning oskusi
- Programmeeritakse erinevates keeltes – Java, C-keel, Python, C++ keel ja teised veel
- Seda õpitakse ülikoolis ja ka mõnes keskkoolis
- Kuid sellega võib ka algust teha juba põhikoolis
- Alustada saab lihtsate asjadega ja liikuda raskemate juurde
- Programmeerida saab mängu, veebilehti ja palju muud!
- Ka päris LEGO roboteid programmeeritakse

20

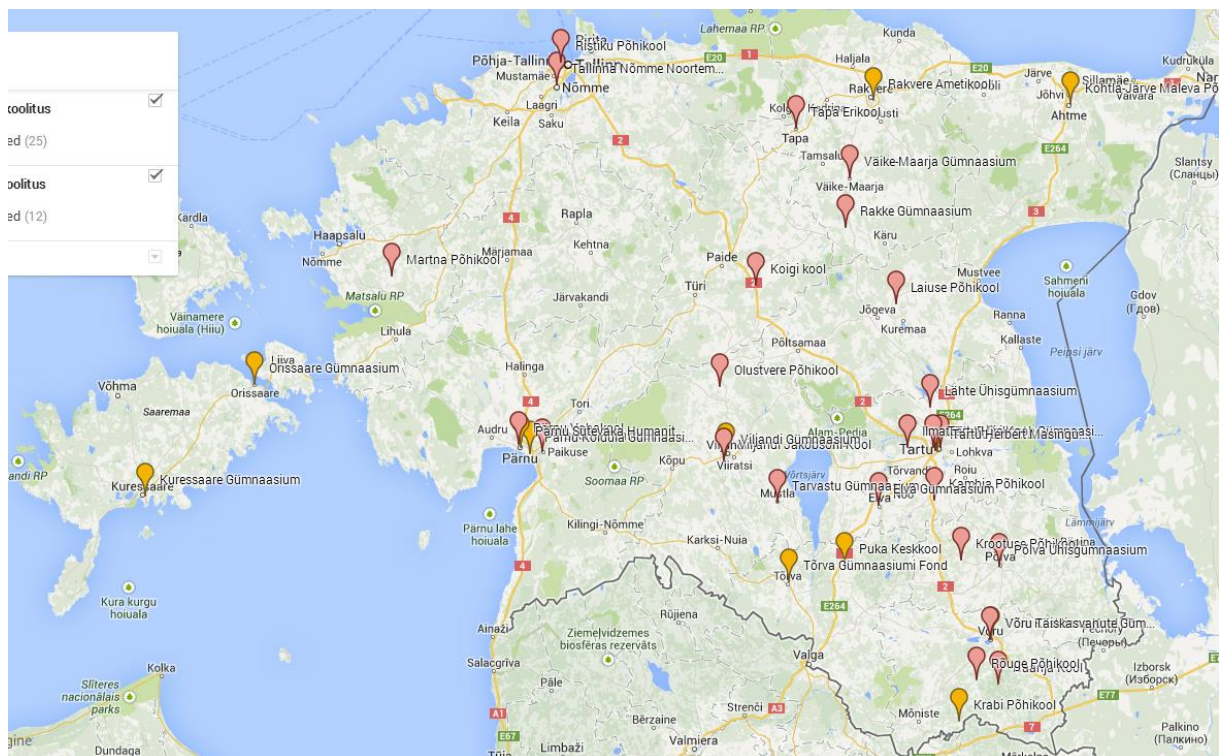


Kuid nüüd küsimuste juurde

- Koostasid need töölehed Sulle õppimiseks
- Nüüd sooviksin teada, kuidas õppimine läks
- Selleks, et küsimustele vastata, suundu palun järgmisele veebilehele:
<http://www.eformular.com/kaitoots/programmeerimine.html>

21

Lisa 7. Õpetajakoolitustel osalenud õpetajate geograafiline paiknemine Eestis



Joonis 4. Roosaga on märgitud Scratchi kursuse ja oranžiga Pythoni kursuse edukalt lõpetanud õpetajate kodukoolid.

(<https://mapsengine.google.com/map/viewer?mid=zJdIoEpy1gXI.ksxb6aLk7rNw>)

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina Kai Sellin

(sünnikuupäev: 13.02.1989)

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose
RoboMind kui võimalus äratada põhikooli õpetajate ja õpilaste huvi programmeerimise vastu, mille juhendajad on Eno Tõnisson ja Sirje Pihlap,
 - 1.1.reprodutseerimiseks säilitamise ja üldsusele kättesaadavaks tegemise eesmärgil, sealhulgas digitaalarhiivi DSpace-is lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
 - 1.2.üldsusele kättesaadavaks tegemiseks Tartu Ülikooli veebikeskkonna kaudu, sealhulgas digitaalarhiivi DSpace'i kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.
3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tartus 23.05.2014